N° 20 MUNDO ATARI

PUBLICACION ATARI INTERNACIONAL EN ESPAÑOL

ENERO 1989 CHILE $400 ARGENTINA A 12 OTROS PAICES US$2

Coca-Cola

Coke

Es sentir de verdad.





PUBLICACION PARA USUARIOS DE MICROCOMPUTADORES ATARI

Editorial

UN nuevo año se nos avecina, amigos de MUNDOATARI, presentando una serie de desafíos que debemos superar con el propósito de dar crecimiento a nuestra revista.

Antes que nada agradecemos su colaboración personal, al hacer posible que la mayoría de las estrategias aplicadas durante el pasado año se cumplieran satisfactoriamente.

En el transcurso del presente año MUNDOATARI proporcionará a ustedes la posibilidad de ampliar su configuración ATARI, ofreciendo los equipos necesarios, al mejor precio del mercado y con nuestra garantía, que significa respaldo y atención permanentes.

En la línea de software ofreceremos programas originales, producidos por SES Sistema, a precio conveniente para la gran cantidad de usuarios que se incorporan progresivamente a la Unidad de Disco.

En este número aparece incluído el facsímil de la revista número 0, que por razones de tiempo y cierre de la revista de navidad debió ser postergada hasta ahora.

Aprovechamos de informar a nuestros lectores nuestro cambio de domicilio a partir del mes de enero. Por razones de fuerza mayor referidas a la autorización de la patente municipal debemos trasladarnos a un nuevo local en la calle Los Leones (ver aviso en páginas interiores).

Con respecto a la definición de material a introducir en los números futuros, adelantamos algunas primicias: nos dedicaremos a profundizar algunos temas específicos como: sonido, color, player missils y a continuar el estudio de lenguajes ASSEMBLER y el FORTH, que iniciamos a partir de este mes.

La estrategia de suscripciones toma un nuevo impulso al favorecer a todos ustedes con la posibilidad de adquirir la revista por la mitad de su valor en kioskos.

Esperando que estos anticipos satisfagan sus necesidades e incrementen nuesta permanente amistad, esperamos su respuesta y colaboración en este nuevo año que se inicia.

Su editor
Iván Gjurovic



ENERO 1989 Precio $ 400

Revista con información exclusiva para microcomputadores ATARI

Resol. Exenta No. 360/6-5-1987

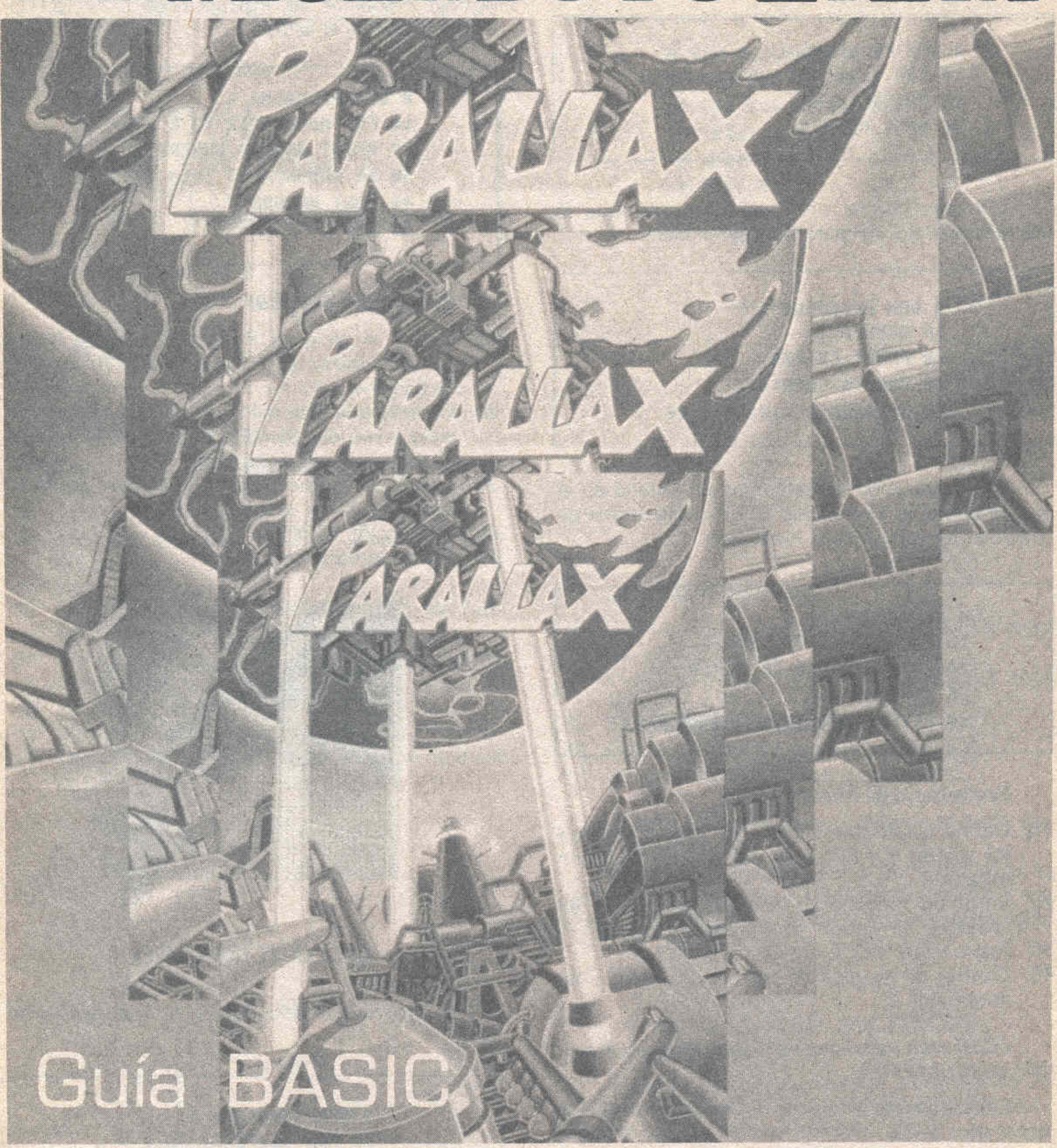
Editor: Iván Gjurovic M.
Director: Adolfo Torrejón S.
Representante legal: Lucía Segura G.
Producción: SES Sistema
Diseño publicitario: Ricardo Numi
Casilla: 458-11, Ñuñoa, Santiago
Teléfono: 2320557

Impresa por EDITORIAL ANTARTICA quien actúa sólo como impresora.

Esta revista no mantiene relación de dependencia de ningún tipo con respecto a los fabricantes de microcomputadores ATARI ni sus representantes.

El contenido de la publicidad es responsabilidad de los avisadores.

MANEJANDO TU ATARI



Este mes continuaremos con el desarrollo de la instrucción FOR/NEXT por su increible versatilidad para controlar una repetición.

Tal como lo mencionamos en el número anterior el ciclo FOR/NEXT consta de dos partes:

- FOR...TO: inicialización con la opción de la definición del rango (STEP).
- NEXT: control de término del ciclo.

En el uso de un ciclo es preferible usar la instrucción FOR/NEXT por varias razones:

1. Requiere menos memoria que un ciclo controlado con IF... THEN.
2. Su ejecución es más rápida.
3. Es más fácil de modificar por los programadores iniciales.

Para corroborar lo afirmado digite y ejecute el siguiente listado:

```
0 REM PROGRAMA 1
10 FOR CUENTA=1 TO 10 STEP 1
20 PRINT CUENTA
30 NEXT CUENTA
```

Compare con el próximo listado en que se usa la instrucción IF para finalizar el ciclo:

```
0 REM PROGRAMA 2
10 CUENTA=1
20 PRINT CUENTA
30 CUENTA=CUENTA+1
40 IF CUENTA<=10 THEN 20
```

A continuación presentamos un listado en el que se añade la instrucción STEP como forma de controlar los rangos de la repetición:

```
0 REM PROGRAMA 3
10 FOR X=1 TO 20 STEP 3
20 PRINT "VALOR DE X ";X
30 NEXT X
40 PRINT "VALOR FINAL DE X ";X
```

Comentario de líneas:

10 Define la variable control de ciclo (X) entre los valores 1 y 20. A su vez define STEP con el valor 3.
20 Edita un mensaje en pantalla para dejar en evidencia la acción de la instrucción STEP.
30 Instrucción NEXT.
40 Control de término de ciclo. Edita valor final de variable X.

¿Cuál cree usted que es el valor final de X editado en la línea 40?

Como consecuencia de la instrucción STEP dicho valor es 22.

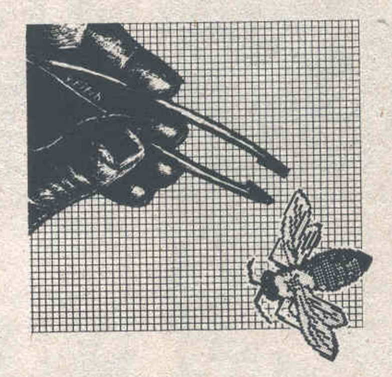


El siguiente listado permite observar la posibilidad de abortar un ciclo antes de su término normal mediante las instrucciones IF y GOTO. Digite y ejecute el listado:

```
0 REM PROGRAMA 4
10 FOR NUMERO=11 TO 5 STEP -1
20 PRINT NUMERO,NUMERO/2
30 IF NUMERO/2<5 THEN GOTO 50
40 NEXT NUMERO
50 END
```

Comentario de líneas:

10 Define el ciclo FOR para la variable NUM. El valor inicial es 11 y el final es 5. El valor de STEP es −1.
20 Observe que esta manera se puede ejecutar un ciclo invertido. Se edita el valor de la variable NUM y la mitad de su valor.
30 Define la condición límite del valor de NUM para finalizar el ciclo.
40 Instrucción NEXT.
50 Fin del programa.



CICLO ILIMITADO

El próximo listado impide la finalización del ciclo:

```
0 REM PROGRAMA 5
10 FOR SINFIN=1 TO 2
20 SINFIN=0
30 NEXT SINFIN
```

Comentario de líneas:

10 Define el ciclo para variable SINFIN entre valores 1 y 2.
20 Asigna valor 0 a variable SINFIN.
30 Instrucción NEXT.

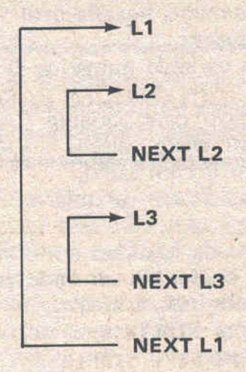
CICLOS ANIDADOS

El siguiente listado ilustra 3 ciclos. Dos de ellos son internos y muestran la estructura de control de flujo:

```
0 REM PROGRAMA 6
10 FOR L1=1 TO 5
20 PRINT "LOOP EXTERIOR :";L1
30 FOR L2=1 TO 5
40 PRINT "LOOP INTERIOR UNO :";L2
50 NEXT L2
60 FOR L3=1 TO 5
70 PRINT "LOOP EXTERIOR DOS :";L3
80 NEXT L3
90 NEXT L1
```

Comentario de líneas:

10 Define el ciclo externo para la variable L1 con los valores entre 1 y 5.
20 Edita el valor de L1.
30 Define el ciclo para variable L2 entre los valores 1 y 5.
40 Edita el valor de L2.
50 Instrucción NEXT para variable L2 definida en línea 30.
60 Define ciclo para variable L3 entre los valores 1 y 5.
70 Edita valor de L3.
80 Finaliza el ciclo L3 inicializado en línea 60.
90 Instrucción NEXT para ciclo externo definido en línea 10.

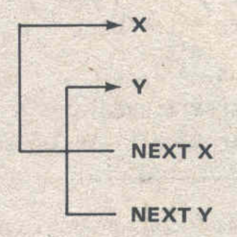


Ejecute el programa y verá como los ciclos internos se repiten las veces definidas en la línea 10 (5 veces).

Modifique los valores de línea 10 para comprobar lo afirmado.

El próximo listado indica la forma de un ciclo anidado con defectos en su construcción:

```
0 REM PROGRAMA 7
10 FOR X=1 TO 10
20 FOR Y=2 TO 20 STEP 2
30 PRINT X,Y
40 NEXT X
50 NEXT Y
```



Comentario de líneas:

10 Define el ciclo externo para variable X entre los valores 1 y 10.
20 Define el ciclo interno para variable Y entre valores 2 y 20 con rango STEP 2.
30 Edita valores de X e Y.
40 Instrucción NEXT para ciclo externo.
50 Instrucción NEXT para ciclo interno.

Ejecute este programa y observe el error producido en pantalla.

Modifique las líneas 40 y 50 cambiando las variables, es decir, en la línea 40 la variable Y que corresponde al ciclo interno y en línea 50 la variable X del ciclo externo.

INICIALIZAR MATRICES

El siguiente listado muestra el uso de la instrucción FOR/NEXT en ciclos anidados como recurso para inicializar matrices.

```
0 REM PROGRAMA 8
10 DIM MATRIZ(10,20)
20 FOR X=0 TO 10
30 FOR Y=0 TO 20
40 MATRIZ(X,Y)=0
50 NEXT Y
60 NEXT X
```

Comentario de líneas:

10 Inicializa o dimensiona la matriz M en un formato de 11 x 21.
20 Define el ciclo para variable X entre valores 0 y 10.
30 Define el ciclo para variable Y entre valores 0 y 20.
40 Inicializa las localizaciones de memoria asignadas para cada elemento de la matriz con el valor 0.
50 Instrucción NEXT para ciclo interno Y.
60 Instrucción NEXT para ciclo externo X.

Ejecute este programa y digite a continuación en modo directo:

? M(5, 5)

El valor resultante será 0.

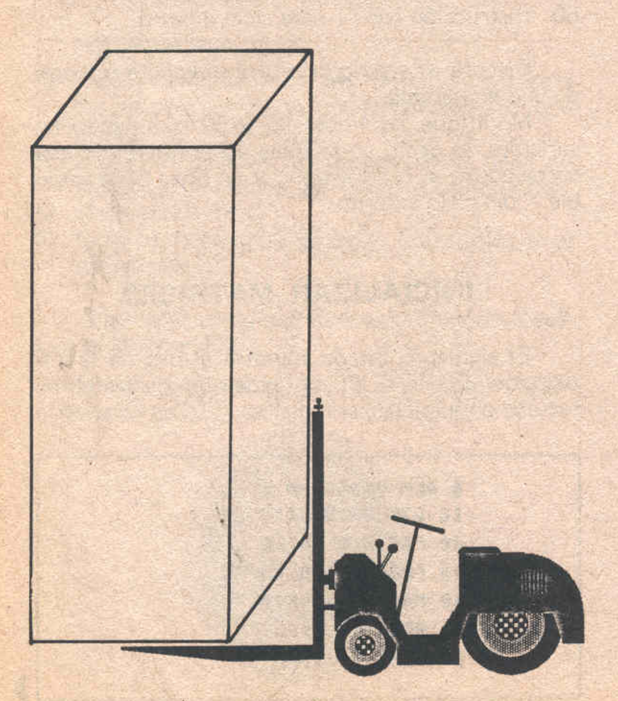
¿Qué importancia tiene la inicialización de una matriz antes de utilizarla?

La importancia reside en la seguridad de que las matrices estén limpias antes de ingresar cualquier valor.

Digite en modo directo:

DIM M(5, 5): ? M(1, 1)

Los valores que se obtienen en esta forma sin inicializar no tienen ningún sentido computacional. Si el valor es igual a 0 es sólo consecuencia de no haber usado la memoria para otro fin, o bien que el computador ha sido recién encendido.



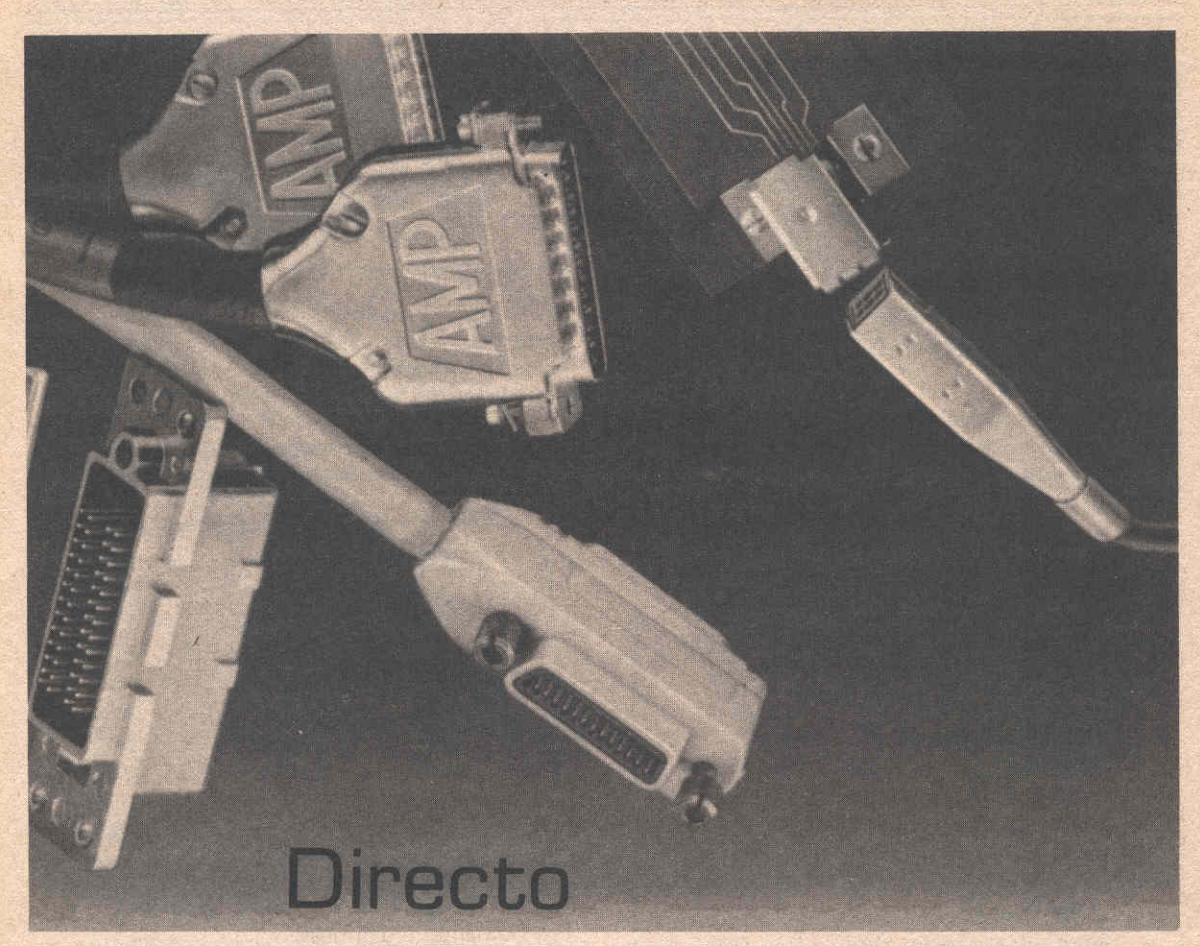
APLICACION

El siguiente listado es un utilitario para convertir números hexadecimales en decimales utilizando la instrucción FOR/NEXT:

```
0 REM PROGRAMA 9
60 DIM HEX$(16),IN$(16)
70 HEX$="0123456789ABCDEF":? CHR$(125)
80 ? :? "INGRESE EL VALOR EN HEXADECIM
AL":INPUT IN$:IF IN$="" OR LEN(IN$)>9
THEN 80
90 DECIMAL=0
100 FOR CAR=1 TO LEN(IN$)
110 FOR VER=1 TO 16
120 IF IN$(CAR,CAR)=HEX$(VER,VER) THEN
 160
130 NEXT VER
140 TRAP 150:IN$(CAR,CAR)=CHR$(ASC(IN$
(CAR,CAR))+128)
150 ? :? "ERROR EN INGRESO ";IN$:TRAP
40000:GOTO 80
160 DECIMAL=DECIMAL*16+(VER-1)
170 NEXT CAR
180 ? " EL VALOR DECIMAL ES ";DECIMAL:
? :GOTO 80
```

Comentario de líneas:

60 Define variables alfanuméricas.
70 Define números de sistema numérico hexadecimal en la variable HEX$.
80 Ingresa número hexadecimal y define condiciones de error.
90 Inicializa variable numérica DEC (valor decimal).
100 Define ciclo para variable CAR (número de caracteres) entre valores 1 y el largo del ingreso (instrucción LEN).
110 Define ciclo para variable VER entre valores 1 y 16.
120 Condición de igualdad entre el valor ingresado y el contenido en variable alfanumérica HEX$ para finalizar ciclo.
130 Instrucción NEXT para ciclo interno VER.
140 Elimina error.
150 Mensaje de error.
160 Asigna valor del dígito hexadecimal en la variable numérica DEC.
170 Instrucción NEXT para ciclo interno de variable CAR.
180 Edita valor decimal.



Directo al 6502

Continuando con la ejercitación del ASSEMBLER presentamos en esta ocasión un programa demostrativo de las capacidades del lenguaje de máquina.

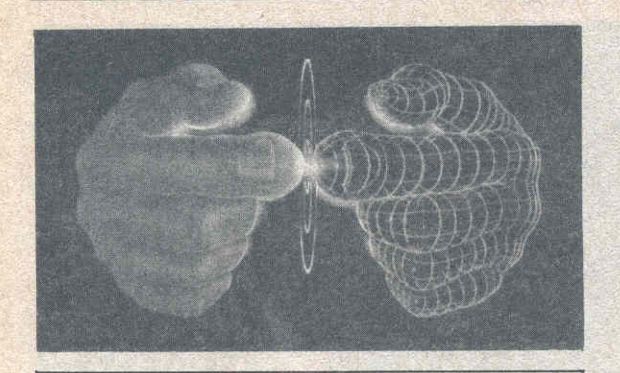
DIBUJO es una rutina para dibujar áreas y llenarlas con un color dado. Es indispensable usar el ASSEMBLER Editor, promocionado por MUNDO-ATARI en su Catálogo.

El programa opera en modo gráfico 3 y llena un cuadro en la parte superior de la pantalla, permitiendo modificarlo para otros modos gráficos o para modificar posiciones en la pantalla.

Digite el programa comprobando que no existan errores. A continuación salve el programa en diskette o casete con los formatos:

LIST #D:DIBUJO o bien LIST #C:

```
19 ; REGISTROS DE COLOR
20 COLPF0 = $2C4
21 COLPF1 = $2C5
22 COLPF2 = $2C6
23 COLPF3 = $2C7
24 COLPF4 = $2C8
29 ; VALORES PARA COLORES
30 ORO    = $18
31 ROJO   = $34
32 AZUL   = $78
33 VERDE  = $C4
39 ; REGISTROS DEL CURSOR
40 ROWCRS = $54
41 COLCRS = $55
42 NEWROW = $60
43 NEWCOL = $61
44 CRSINH = $02F0
45 OLDROW = $5A
46 OLDCOL = $5B
47 FILDAT = $2FD
48 ATACHR = $2FB
```

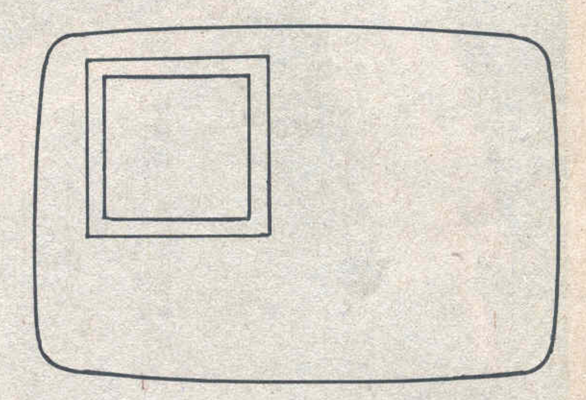


```
99 ; PROGRAMA CENTRAL
0100 *=$600
0110 DEV .BYTE "S:"
0120  LDX #$60
0130  LDA #$3
0140  STA $342,X
0150  LDA #DEV&$00FF
0160  STA $344,X
0170  LDA #DEV/256
0180  STA $345,X
0190  LDA #$0C
0200  STA $34A,X
0210  LDA #3
0220  STA $34B,X
0230  JSR $E456
0240  LDA #ROJO
0250  STA COLPF1
0260  LDA #2
0270  STA ATACHR
0280  LDA #0
0290  STA OLDCOL
0300  LDA #12
0310  STA OLDROW
0320  STA COLCRS
0330  STA ROWCRS
0340  LDX #$60
0350  LDA #$11
0360  STA $342,X
0370  JSR $E456
0380  LDA #0
0390  STA ROWCRS
0400  JSR $E456
0410  LDA #0
0420  STA COLCRS
0430  JSR $E456
0440  LDA #12
0450  STA ROWCRS
0460  JSR $E456
0470  LDA #0
0480  STA ROWCRS
0490  LDA #1
```

```
0500  STA FILDAT
0510  LDX #$60
0520  LDA #$12
0530  STA $342,X
0540  JSR $E456
0550 STOP JMP STOP
```

Digite la instrucción ASM y al cabo de un momento aparecerá en pantalla el listado ensamblado del programa objeto con su correspondiente ubicación en memoria a partir de la página 6.

Para ejecutar digite la instrucción BUG y luego G600, que corresponde al inicio del programa. En pantalla aparece la imagen siguiente:

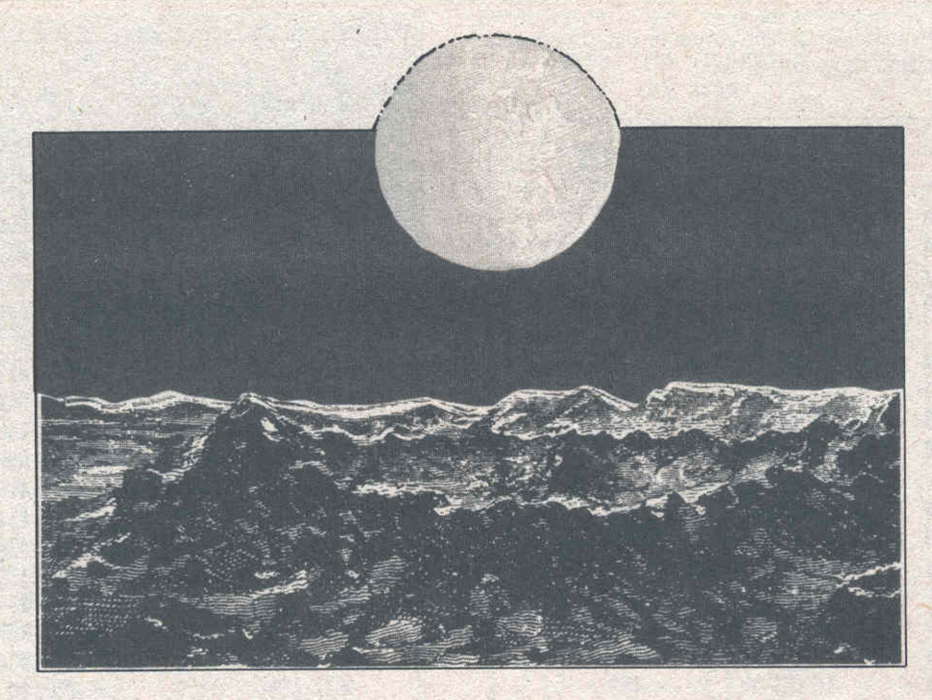


Comentario de líneas:

20-24 Define localizaciones para registros en color en hexadecimal.
30-33 Proporciona valores para los colores.
40-48 Define localizaciones de registros del cursor.
100 Operador que define inicio del programa objeto.
110-230 Abre el canal No. 6 y activa modo gráfico 3.
240-250 Carga el color rojo en el registro $2C4.
260-270 Pone valor 2 en la localización ATACHR ($2FB ó decimal 763).
280 Carga acumulador con valor 0.
290 Almacena valor en OLDCOL (columna anterior).
300 Carga acumulador con valor 12.
310 Almacena valor en la localización OLDROW.
320-330 Dibuja puntos para comando DRAWTO.
350-360 Inicializa byte para comando DRAW ($11 ó decimal 17) y lo almacena en la localización ICCOM.
370 Ejecuta la rutina del Sistema Operativo.
380-460 Completa el dibujo del cuadrado.
470-480 Localiza el cursor en la posición superior izquierda de la pantalla.
490-540 Llena el color especificado usando la rutina localizada en $E456.

A continuación presentamos el listado ensamblado tal como se ve en pantalla:

```
              19 ; REGISTROS DE COLOR
02C4          20 COLPF0  =   $2C4
02C5          21 COLPF1  =   $2C5
02C6          22 COLPF2  =   $2C6
02C7          23 COLPF3  =   $2C7
02C8          24 COLPF4  =   $2C8
              29 ; VALORES PARA COLORES
0018          30 ORO     =   $18
0034          31 ROJO    =   $34
0078          32 AZUL    =   $78
00C4          33 VERDE   =   $C4
              39 ; REGISTROS DEL CURSOR
0054          40 ROWCRS  =   $54
0055          41 COLCRS  =   $55
0060          42 NEWROW  =   $60
0061          43 NEWCOL  =   $61
02F0          44 CRSINH  =   $02F0
005A          45 OLDROW  =   $5A
005B          46 OLDCOL  =   $5B
02FD          47 FILDAT  =   $2FD
02FB          48 ATACHR  =   $2FB
              99 ; PROGRAMA CENTRAL
0000          0100    *=     $600
0600 53       0110 DEV    .BYTE "S:"
0601 3A
0602 A260     0120        LDX  #$60
0604 A903     0130        LDA  #$3
0606 9D4203   0140        STA  $342,X
0609 A900     0150        LDA  #DEV&$00FF
060B 9D4403   0160        STA  $344,X
060E A906     0170        LDA  #DEV/256
0610 9D4503   0180        STA  $345,X
0613 A90C     0190        LDA  #$0C
0615 9D4A03   0200        STA  $34A,X
0618 A903     0210        LDA  #3
061A 9D4B03   0220        STA  $34B,X
061D 2056E4   0230        JSR  $E456
0620 A934     0240        LDA  #ROJO
0622 8DC502   0250        STA  COLPF1
0625 A902     0260        LDA  #2
0627 8DFB02   0270        STA  ATACHR
062A A900     0280        LDA  #0
062C 855B     0290        STA  OLDCOL
062E A90C     0300        LDA  #12
0630 855A     0310        STA  OLDROW
0632 8555     0320        STA  COLCRS
0634 8554     0330        STA  ROWCRS
0636 A260     0340        LDX  #$60
0638 A911     0350        LDA  #$11
063A 9D4203   0360        STA  $342,X
063D 2056E4   0370        JSR  $E456
0640 A900     0380        LDA  #0
0642 8554     0390        STA  ROWCRS
0644 2056E4   0400        JSR  $E456
0647 A900     0410        LDA  #0
0649 8555     0420        STA  COLCRS
064B 2056E4   0430        JSR  $E456
064E A90C     0440        LDA  #12
0650 8554     0450        STA  ROWCRS
0652 2056E4   0460        JSR  $E456
0655 A900     0470        LDA  #0
0657 8554     0480        STA  ROWCRS
0659 A901     0490        LDA  #1
065B 8DFD02   0500        STA  FILDAT
065E A260     0510        LDX  #$60
0660 A912     0520        LDA  #$12
0662 9D4203   0530        STA  $342,X
0665 2056E4   0540        JSR  $E456
0668 4C6806   0550 STOP   JMP  STOP
```



USR

La columna USR da cabida en este número a una interesante colaboración enviada por un lector de Arica, don Carlos Hasan V., quien ha investigado en el movimiento de scrolling fino.

Su rutina permite mover las localizaciones de la memoria ROM por medio del joystick.

Nuestro lector recibirá un premio de $ 3.000 en software a eleccción de nuestro Catálogo.

Respecto a su consulta de utilizar más de 4 player, es posible sólo aumentar un quinto player sacrificando los 4 missiles.

```
1 REM Demostracion scrolling
2 REM CARLOS HASAN V.
3 REM **Mover con el joystick**
4 REM
10 GRAPHICS 0:? "ESPERE UN MOMENTO..."
15 FOR X=0 TO 240:READ A:POKE 1536+X,A
:NEXT X
20 A=40120:POKE 205,A-256*INT(A/256):P
OKE 206,INT(A/256)
25 Z=USR(1710)
100 DATA 173,120,2,41,248,208,26,198,2
03,165,203,201,255,208,18,169,3,133,20
3,230,205,165,205,201,217
110 DATA 208,6,169,0,133,203,198,205,1
73,120,2,41,244,208,26,230,203,165,203
,201,4,208,18,169,0
120 DATA 133,203,198,205,165,205,201,2
55,208,6,169,3,133,203,230,205,162,69,
165,205,157,4,1,202,202
130 DATA 202,16,248,173,120,2,41,241,2
08,26,198,204,165,204,201,255,208,18,1
69,7,133,204,198,206,165
140 DATA 206,201,255,208,6,169,0,133,2
04,230,106,173,120,2,41,242,208,26,230
,204,165,204,201,8,208
150 DATA 18,169,0,133,204,230,206,165,
206,201,233,208,6,169,7,133,204,198,20
6,162,0,165,206,157,5
160 DATA 1,24,105,1,232,232,232,224,72
,208,243,165,203,141,4,212,165,204,141
,5,212,76,98,228,104
170 DATA 162,2,169,112,157,0,1,202,16,
250,162,66,169,114,157,3,1,202,202,202
,16,248,169,82,141
180 DATA 72,1,169,65,141,75,1,169,0,14
1,76,1,141,48,2,133,203,133,204,169,1,
141,77,1,141
190 DATA 49,2,169,35,141,47,2,160,0,16
2,6,169,7,76,92,228
200 END
```

A continuación presentamos la versión ensamblada del programa en función USR tratado en la página 27. Los comentarios en inglés son fáciles de comprender .

```
0160 ; System equates
0170 ;
0180 SAVMSC = $58      ; Addr of screen
0190 BASRET = $D4      ; BASIC return
0200 RTCLOK = $13      ; System timer
0210 ;
0220 ; For simplicity, we'll let
0230 ; BASIC set up GRAPHICS 24.
0240 ;
0250      PLA          ; # args; ignore
0260 ;
0270 ; Zero the system timers
0280 ;
0290      LDA #0
0300      STA RTCLOK
0310      STA RTCLOK+1
0320 ;
0330 ; This is the screen-fill loop
0340 ;
0350        LDX #32         ; Loop 32 times
0360 NEXTI
0370        LDY #0          ; Init J-loop
0380        LDA #255        ; Fill byte
0390 NEXTJ
0400        STA (SAVMSC),Y ; Poke it!
0410        INY
0420        CPY #240        ; 240 pokes yet?
0430        BNE NEXTJ       ; No; do another
0440        CLC
0450        LDA SAVMSC      ; Add 240 to
0460        ADC #240        ; screen addr
0470        STA SAVMSC
0480        LDA SAVMSC+1 ; This is a
0490        ADC #0          ; 16-bit add
0500        STA SAVMSC+1
0510        DEX             ; 32 loops yet?
0520        BNE NEXTI       ; No; do another
0530 ;
0540 ; Screen is now filled, so pass
0550 ; the timer readings to BASIC
0560 ;
0570        LDA RTCLOK
0580        STA BASRET+1
0590        LDA RTCLOK+1
0600        STA BASRET
0610        RTS             ; Back to BASIC!
```



Correo

MUNDOATARI inicia a partir de esta edición una columna destinada a acoger la numerosa correspondencia recibida en este último tiempo. Por razones de espacio nos reservamos el derecho de resumir las cartas publicadas, sin perder el sentido original de ellas.

DEFENSA ESTELAR

En el número 14 de MUNDOATARI publicaron ustedes un programa para tipiar.

En la línea 900 surge un error que se detecta en el símbolo de los dos puntos (:) que separa los tipos VR: FOR error que hemos tratado de solucionar efectuando cambios de simbología, sin resultados aceptables.

Agradeceré su aclaración al respecto, para grabarlo para deleite de mis hijos, fanáticos de estos programas.

Guillermo Barriga B.
Ill. Municipalidad de Coihueco
Prat No. 683, Coihueco

☐ *En realidad no existe tal error. VR es el nombre de una variable que se asocia al comando READ. Por otro lado la instrucción FOR... corresponde al estamento siguiente. Los dos puntos separan los comandos anteriores.*

Desgraciadamente no tenemos mayor información acerca del error producido, el que podría encontrarse también en los DATA de la línea 920.

M.A.

MUNDOATARI EN BOLIVIA

Somos un grupo de profesionales con el hobby de los computadores ATARI. Al ver su capacidad como elemento formativo y de trascendencia cultural hemos decidido formar una sociedad y empezar en nuestro medio el trabajo que está tan avanzado en vuestro país.

Hemos elaborado un directorio de cerca de 5.000 usuarios ATARI.

Solicitamos información acerca de suscripciones a MUNDOATARI y de software educativo de la empresa SES Sistema.

Estamos dispuestos a colaborar con ustedes para que en un futuro podamos unir esfuerzos y acrecentar la gran familia latinoamericana ATARI.

Lic. Franklin Poppe A.
Miguel A. Morales
La Paz, Bolivia

MUNDOATARI EN COLOMBIA

Una empresa representante de ATARI en Colombia llevó como primicia los primeros números de MUNDOATARI, pero su despacho se suspendió a partir del cuarto número.

Me gustaría entablar comunicación con ustedes con el fin de suscribirme a su revista, que nos parece una guía maravillosa para quienes poseemos equipos ATARI.

Carlos Cárdenas P.
A.A. No.2481, Villavicencio-Meta
Colombia

Dominando la 1050

Este mes trataremos las posibilidades de traspaso entre la memoria (área de pantalla) y la diskettera a través de un interesante utilitario.

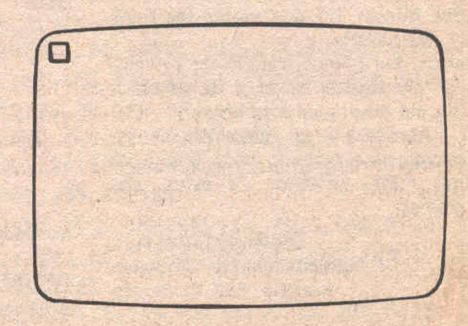
Este programa es una alternativa para crear pantallas usando el set de caracteres disponible en el teclado, editando una determinada pantalla y luego salvarla en un diskette para uso posterior.

La estrategia de este programa es mediante el IOCB (Input/Output Control Block) y el CIO (Central I/O).

La siguiente tabla proporciona las localizaciones de los IOCB correspondientes:

IOCB	ICCOM	ICBLL	ICBLH	ICBAL	ICBAH
1	850	856	857	852	853
2	866	872	873	868	869
3	882	888	889	884	885
4	898	904	905	900	901
5	914	920	921	922	923
6	Prioridad a pantalla (S:)				
7	Prioridad a LPRINT y LOAD				

ICCOM : contiene la operación a realizar (7: leer, 11: escribir).
ICBLL : byte menor de la cantidad de bytes a mover.
ICBLH : byte mayor de la cantidad de bytes a mover.
ICBAL : byte menor de la primera localización de memoria de pantalla.
ICBAH : byte mayor de la primera localización de memoria de pantalla.



El programa contiene una rutina en lenguaje de máquina con la función USR, que permite el traspaso rápido de caracteres.

```
10 DIM F$(16),ML$(7)
20 FOR X=1 TO 7:READ A:ML$(X)=CHR$(A):
NEXT X
30 DATA 104,104,104,170,76,86,228
40 PRINT CHR$(125)
50 TRAP 400:OPEN #2,4,0,"K:"
60 GET #2,KC:IF KC=155 THEN DIR=11:IO=
8:GOTO 110
70 IF KC=27 THEN DIR=7:IO=4:GOTO 100
90 ? CHR$(KC);:GOTO 60
100 POSITION 2,21:? " DIGITE NOMBRE AR
CHIVO QUE CARGA ":INPUT F$:GOTO 300
110 POSITION 2,21:? " DIGITE NOMBRE AR
CHIVO QUE SALVA ":INPUT F$
300 OPEN #1,IO,0,F$
310 SCREEN=PEEK(88)+256*PEEK(89)
320 STARTHB=INT(SCREEN/256)
330 STARTLB=SCREEN-256*STARTHB
340 POKE 852,STARTLB:POKE 853,STARTHB
350 POKE 856,31:POKE 857,3
360 POKE 850,DIR:A=USR(ADR(ML$),16)
370 CLOSE #1
380 POSITION 2,21:? "
                    "
385 POSITION 2,22:? "
                    ";
390 GOTO 60
400 POSITION 2,21:? "
                    "
405 POSITION 2,22:? "
                    ";
420 IF PEEK(195)=138 THEN POSITION 2,2
1:? " DRIVE NO DISPONIBLE ":FOR PAUSA=
1 TO 500:NEXT PAUSA:GOTO 370
430 IF PEEK(195)=170 THEN POSITION 2,2
1:? " ARCHIVO NO SE ENCUENTRA ":FOR PA
USA=1 TO 500:NEXT PAUSA
440 GOTO 370
```

Al ejecutar el listado la pantalla se vuelve blanca, excepto el cursor. Usted tiene control sobre las teclas para digitar lo que desee, incluyendo borrar caracteres, insertar líneas, caracteres gráficos, inversos y símbolos.

Se necesitan 2 comandos para efectuar el traspaso: la tecla RETURN se usa para salvar la pantalla, indicando el nombre del archivo con el formato "D: nombre.".

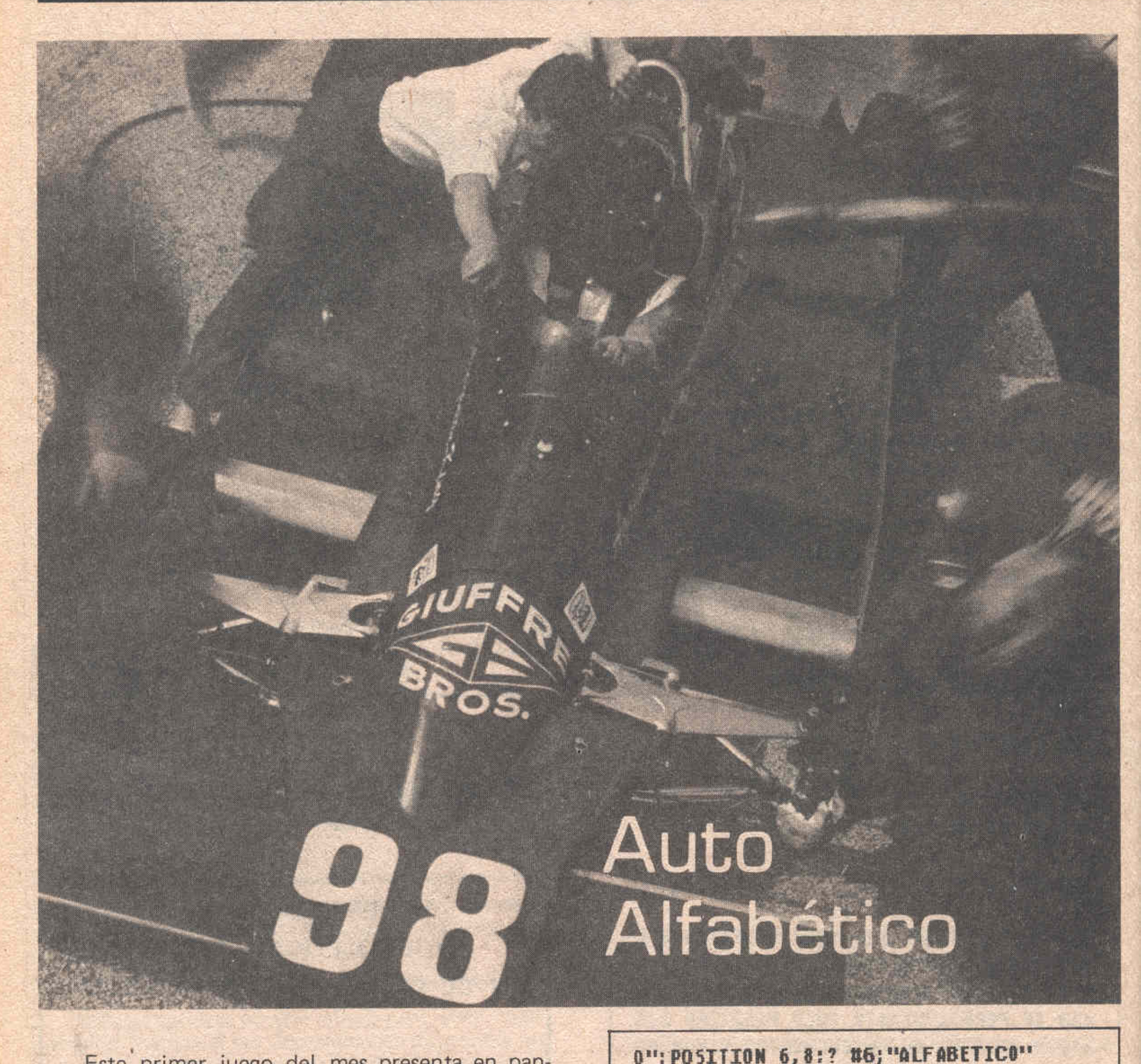
La tecla ESC se usa para cargar la pantalla indicando el nombre del archivo.

Cada pantalla está en modo 0 y utiliza sólo 7 sectores de almacenamiento del disco.



Comentario de líneas:

10-50 Inicialización.
60-90 Controla los caracteres desde el teclado, condicionando las teclas RETURN y ESC.
100-110 Pide el nombre del archivo (pantalla).
300-370 Inicializa valores para CIO usando el IOCB #1.
380-390 Limpia la pantalla.
400-440 Rutinas de error. ●



Este primer juego del mes presenta en pantalla un auto, varias letras y algunos bloques distribuidos al azar.

Por medio de las teclas de las flechas se puede dirigir el movimiento del auto para hacer desaparecer las letras en orden alfabético.

Es un simpático juego con animación en BASIC y útil para aprender la técnica.

```
100 CAR=3;PAN=1:SC=0:MAX=0:OBS=5:MAN=3
:OLD=193
110 MAX=MAX+5:OBS=OBS+5
120 LIM=MAX
130 SET=(PEEK(106)-8)*256:IF PEEK(SET+
8)=7 THEN 170
140 GRAPHICS 18:POSITION 8,6:? #6;"AUT
O":POSITION 6,8:? #6;"ALFABETICO"
150 FOR T=0 TO 511:POKE SET+T,PEEK(573
44+T):NEXT T
160 FOR T=0 TO 55:READ N:POKE SET+T,N:
NEXT T
161 DATA 0,0,0,0,0,0,0,0
162 DATA 7,119,34,254,254,34,119,7
163 DATA 224,238,68,127,127,68,238,224
164 DATA 24,90,126,90,24,219,255,195
165 DATA 195,255,219,24,90,126,90,24
166 DATA 0,126,126,126,126,126,126,0
167 DATA 255,255,255,255,255,255,255,2
55
170 GRAPHICS 17:POKE 756,SET/256
180 POSITION 1,0:? #6;"AUTO - ALFABETI
CO":COLOR 38:PLOT 0,1:DRAWTO 19,1
190 DRAWTO 19,21:DRAWTO 0,21:DRAWTO 0,
```

```
2
200 POSITION 1,22:? #6;"PUNTAJE:";SC:P
OSITION 1,23:? #6;"PANTALLA:";PAN
210 POSITION 13,23:? #6;"AUTO:";CAR
220 FOR T=1 TO OBS
230 GOSUB 390
250 COLOR 133:PLOT C,L
260 NEXT T
270 FOR T=1 TO MAX
280 GOSUB 390
310 LOCATE C,L,Z:IF Z=133 THEN 280
320 COLOR NEO+160:PLOT C,L
330 NEXT T
340 GOSUB 390
370 LOCATE C,L,Z:IF Z>=133 AND Z<=218
THEN 340
380 COLOR MAN:PLOT C,L:GOTO 400
390 L=INT(RND(0)*19)+2:C=INT(RND(0)*18
)+1:NEO=INT(RND(0)*26)+33:RETURN
400 XC=C:XL=L:T=PEEK(764)
410 C=C+(T=7)-(T=6)
420 L=L+(T=15)-(T=14)
430 IF C=XC AND L=XL THEN 400
440 MAN=(T=6)*1+(T=7)*2+(T=14)*3+(T=15
)*4
460 COLOR 0:PLOT XC,XL:LOCATE C,L,P
465 A=1^1^1^1
470 IF P=38 OR P=133 THEN 510
480 IF P>=193 AND P<=218 THEN 540
500 COLOR MAN:PLOT C,L:GOTO 400
510 SOUND 1,72,12,10:FOR T=1 TO 50:NEX
T T:SOUND 1,0,0,0
520 CAR=CAR-1:POSITION 18,23:? #6;CAR:
IF CAR=0 THEN 740
530 FOR T=1 TO 1500:NEXT T:OLD=193:POK
E 764,255:GOTO 120
540 NEO=P:IF NEO<OLD THEN 710
550 SC=SC+50:LIM=LIM-1:OLD=NEO
560 POSITION 9,22:? #6;SC
570 SOUND 1,10,10,10:FOR T=1 TO 50:NEX
T T:SOUND 1,0,0,0
580 IF LIM>0 THEN 400
590 PAN=PAN+1:IF PAN=6 THEN PAN=1:MAX=
0:OBS=5:OLD=193
600 POSITION 1,9:? #6;" CONGRATULACION
ES ":POKE 764,255
610 SOUND 2,72,10,8:GOSUB 690
620 SOUND 2,64,10,8:GOSUB 690
630 SOUND 2,60,10,8:GOSUB 690
640 SOUND 2,72,10,8:GOSUB 690
650 SOUND 2,64,10,8:GOSUB 690
660 SOUND 2,72,10,8:GOSUB 690
670 Q=Q+1:IF Q<11 THEN 610
680 Q=0:GOTO 700
690 FOR T=1 TO 30:NEXT T:SOUND 2,0,0,0
:RETURN
700 POSITION 10,23:? #6;PAN:OLD=193:GO
TO 110
710 POSITION 1,0:? #6;"   VA ANTES QUE
   "
720 COLOR NEO:PLOT 2,0:COLOR OLD:PLOT
17,0
730 GOTO 510
740 POSITION 3,7:? #6;" FIN DE JUEGO "
:POSITION 4,11:? #6;" PRES. SPACE "
750 FOR T=0 TO 10:FOR T1=0 TO 70
755 SOUND 2,T1,10,10-T
760 NEXT T1:NEXT T:SOUND 2,0,0,0
770 IF PEEK(764)<>33 THEN 750
780 RUN
```

Equilibrio

El segundo juego es una colaboración enviada por don Mario Villarroel.

Se trata de un juego de destreza con el joystick para dirigir un símbolo a través de la pantalla. Hay varios niveles de dificultad.

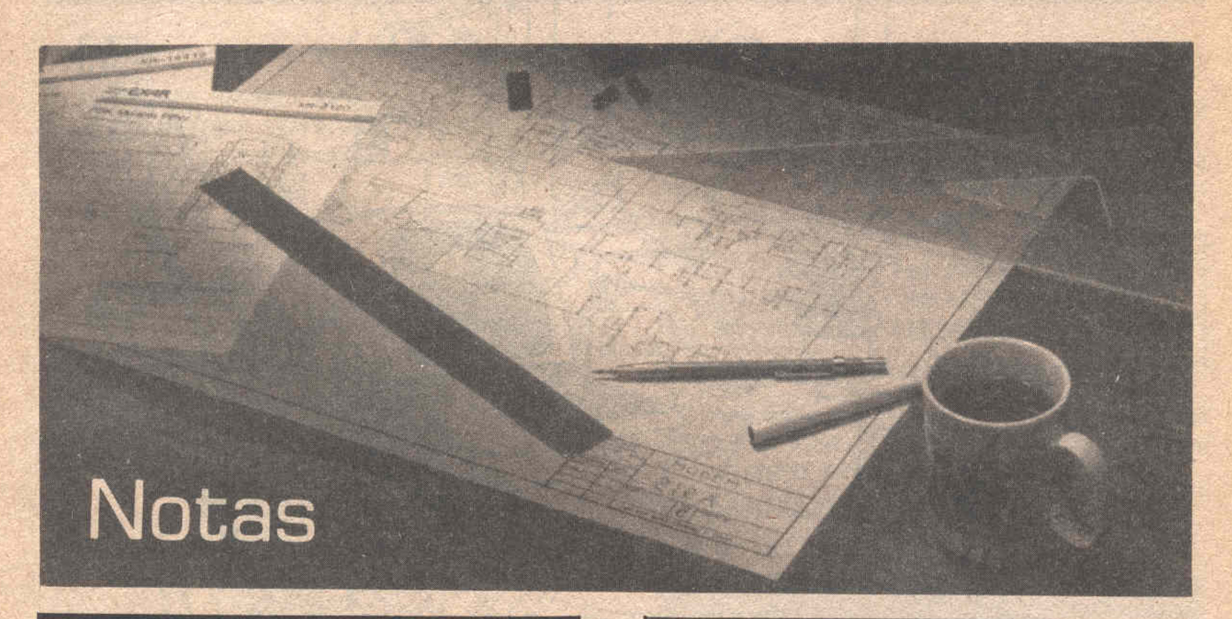
Nuestro colaborador se hace acreedor a un premio de $ 3.000 en software del Catálogo, a elección.

Felicitaciones.

```
0 GRAPHICS 2+16:SETCOLOR 2,0,0:SETCOLO
R 1,8,14:POKE 752,1:POSITION 2,2:? #6;
"EQUILIBRIO":GOSUB 20000
1 DIM M(35,10),Y(35,10),Z(35,10),N$(7)
,A$(1),Q(35,10)
2 OPEN #1,4,0,"K:":S=200
3 K=0:V=3:W=0:P=0:SOUND 0,0,0,0
4 A=0:X=200:W=W+1:SOUND 0,0,0,0
9 D=10:G=0:F=10
10 GRAPHICS 0:SETCOLOR 2,0,0:SETCOLOR
1,8,14:POKE 752,1:SETCOLOR 4,3,12
15 POSITION 0,0:? "                EQUI
LIBRIO            "
20 POSITION 0,1:? "RECORD : ";R:POSITI
```



```
ON 28,1:? "DE: ";N$
25 POSITION 0,2:? "PUNTAJE: ";P
27 POSITION 16,2:? "VIDAS   : ";V
35 FOR T=1 TO 35
38 A=INT(RND(0)*3)
45 M(T,10)=A
50 IF M(T,10)=2 THEN M(T,10)=-1
55 IF T=1 THEN M(1,10)=0
60 G=G+M(T,10):Y(T,10)=G
61 Q(T,10)=Y(T,10)+10
63 IF Q(T,10)<=6 THEN Q(T,10)=6
64 IF Q(T,10)>=18 THEN Q(T,10)=18
65 POSITION 0+T,Q(T,10):? "-"
70 NEXT T
80 FOR T=1 TO 35
85 IF STICK(0)=14 THEN J=-1:GOTO 100
90 IF STICK(0)=13 THEN J=1:GOTO 100
95 IF STICK(0)=15 THEN J=0:GOTO 100
100 F=F+J:Z(T,10)=F
101 IF T=1 THEN Z(T,10)=10
102 Y(T,10)=Y(T,10)+10
105 IF Z(T,10)=Q(T,10) THEN 110
106 IF Z(T,10)<>Q(T,10) THEN 1000
110 POSITION 0+T,Z(T,10):? ">"
120 FOR PAUSE=1 TO 5:NEXT PAUSE
121 SOUND 0,T,10,15:SOUND 0,0,0,0
122 P=P+10:POSITION 9,2:? P
123 IF P>R THEN R=P:POSITION 9,1:? P
124 IF P=5000 THEN GOSUB 15000
130 NEXT T:S=S-10:GOTO 10000
140 END
1000 V=V-1
1001 IF V<0 THEN K=K+1
1002 POSITION 13,18:? "TE DESEQUILIBRASTE "
1003 POSITION 15,19:? "TE CAISTE AL PRECIPICIO":W=W-1
1004 POSITION 17,20:? "MALA SUERTE!!!":FOR T=250 TO 1 STEP -1:SOUND 0,T,10,15:NEXT T:FOR PAUSE=1 TO 250:NEXT PAUSE
1005 IF K=0 THEN 4
1006 IF K>=1 THEN 1007
1007 IF R>P THEN 1100
1008 IF P>R THEN 1010
1010 GRAPHICS 2+16:SETCOLOR 2,0,0:SETCOLOR 1,8,14:POKE 752,1
1012 POSITION 2,4:? #6;"game over"
1020 FOR PAUSE=1 TO 500:NEXT PAUSE
1030 GRAPHICS 0:SETCOLOR 2,0,0:SETCOLOR 4,3,12:SETCOLOR 1,8,14:POKE 752,1
1040 POSITION 6,6:? "SACASTE  EL MEJOR PUNTAJE"
1050 POSITION 6,8:? "ANOTA TU NOMBRE: ";:INPUT N$
1060 POSITION 6,10:? "QUIERES OTRO JUEGO (S/N)?"
1066 GET #1,X:A$=CHR$(X)
1070 IF A$="S" THEN P=0:GOTO 3
1080 IF A$="N" THEN END
1090 GOTO 1066
1100 GRAPHICS 2+16:SETCOLOR 2,0,0:SETCOLOR 1,8,14:POKE 752,1
1110 POSITION 2,4:? #6;"game over"
1120 FOR PAUSE=1 TO 500:NEXT PAUSE
1130 GRAPHICS 0:SETCOLOR 2,0,0:SETCOLOR 4,3,12:SETCOLOR 1,8,14:POKE 752,1
1160 POSITION 6,10:? "QUIERES OTRO JUEGO (S/N)?"
1166 GET #1,X:A$=CHR$(X)
1170 IF A$="S" THEN P=0:GOTO 3
1180 IF A$="N" THEN END
1190 GOTO 1066
10000 GRAPHICS 2+16:SETCOLOR 2,0,0:SETCOLOR 1,8,14:POKE 752,1
10010 POSITION 2,4:? #6;"ETAPA ";W
10020 FOR PAUSE=1 TO 500:NEXT PAUSE
10030 GOTO 4
15000 FOR T=1 TO 200:SOUND 0,T,10,15:NEXT T:SOUND 0,0,0,0
15001 V=V+1:POSITION 25,2:? V
15002 RETURN
20000 POSITION 2,4:? #6;"por"
20001 POSITION 1,7:? #6;"mario villarroel"
20002 FOR PAUSE=1 TO 500:SOUND 0,PAUSE,10,15:NEXT PAUSE:SOUND 0,0,0,0
20003 RETURN
```



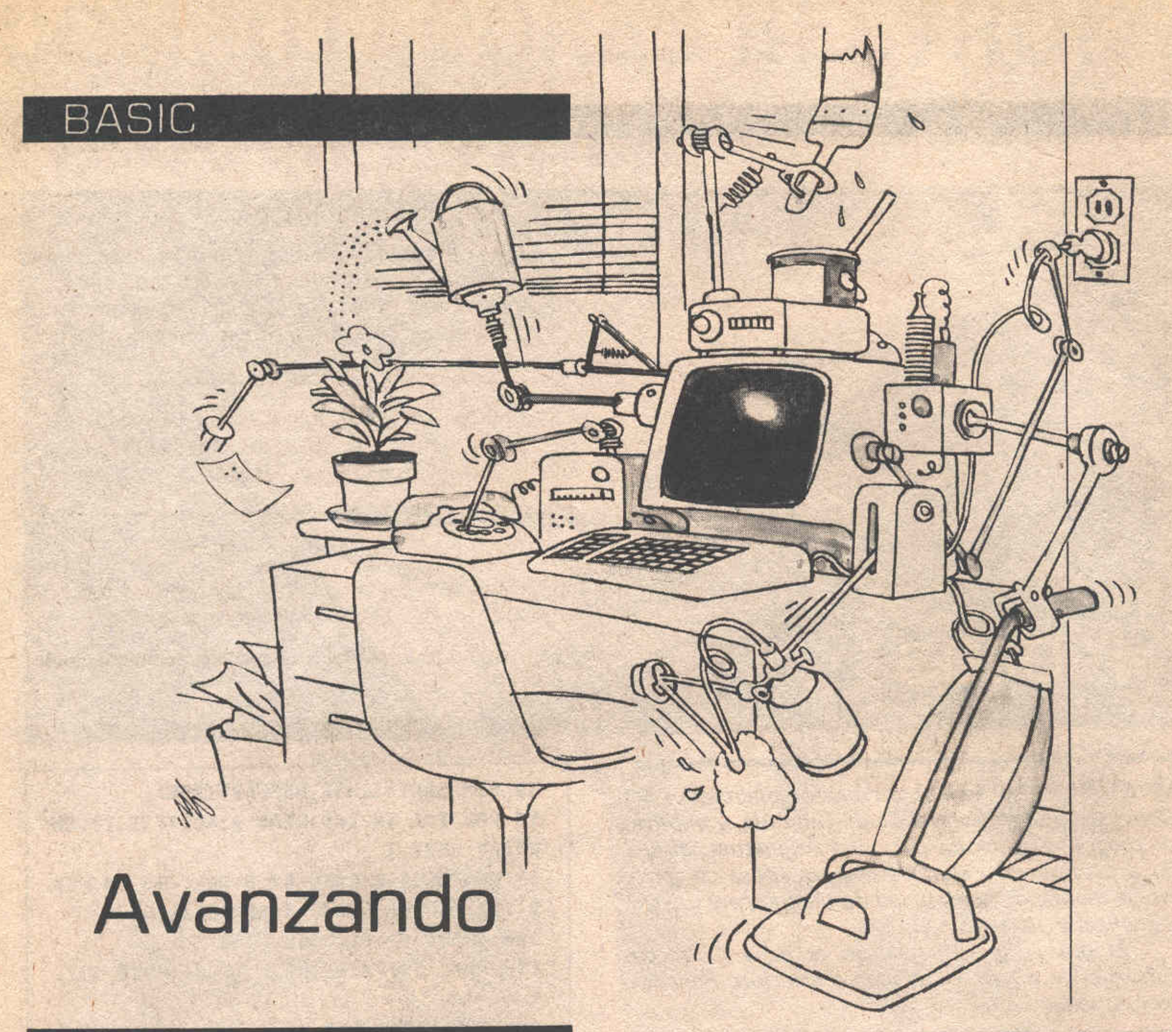
Notas

MUNDOATARI felicita a nuestro colaborador de Concepción, don Gabriel Cerda Quezada, y publica su programa de Notas para una Asignatura, dirigido a profesores y alumnos que en época de vacaciones tienen disponibilidad de tiempo para digitar y analizar el listado adjunto.

El premio por su trabajo es de $ 7.000 en software de nuestro Catálogo a elección. Felicitaciones, señor Cerda.

```
1 REM PROGRAMA DE ARCHIVO PARA UNA
2 REM ASIGNATURA
3 REM GABRIEL CERDA QUEZADA
4 REM SOCIO Nro. 68-575000-5
5 REM
6 REM CONCEPCION
7 REM VIII  REGION
8 REM C H I L E
9 REM * * * * * * * * * * * * * * *
20 REM NOTAS PARA UNA ASIGNATURA
25 DIM ASIG$(16),CUR$(3),NOM$(25),R$(3
),B$(65),ARCH$(6500)
26 I=12:DIM N(I),NX$(3)
40 B$=" ":B$(65)=" ":B$(2)=B$
50 ARCH$=" "
60 GRAPHICS 2:POKE 82,0:POKE 83,38:POK
E 752,1:POKE 53774,64:POKE 16,64
65 POSITION 0,2:? #6;" ARCHIVO DE NOTA
S "
70 ? :? " Creado por: Gabriel Cerda Q.
"
72 ? "                          Enero
 1988 "
90 DIM SORT$(169),BUFFER$(256)
95 FOR I=1 TO 169:READ A:SORT$(I,I)=CH
R$(A):NEXT I
97 GRAPHICS 0:POKE 82,0:POKE 83,38:POK
E 752,1:POKE 53774,64:POKE 16,64
100 GOSUB 5200:GOSUB 6000
102 POKE 53774,64:POKE 16,64:POKE 752,
1
105 CLOSE #1:OPEN #1,4,0,"K:":? CHR$(1
25)
108 POKE 53774,64:POKE 16,64:POKE 752,
1
110 TRAP 110:POSITION 10,3:? " Menu P
rincipal "
115 POSITION 3,5:? "1- Listado"
120 POSITION 3,7:? "2- Ingreso de Dato
s":ZP=0
125 POSITION 3,9:? "3- Modificar Datos
"
130 POSITION 3,11:? "4- Eliminar Perso
nas"
135 POSITION 3,13:? "5- Imprimir"
140 POSITION 3,15:? "6- Ordenar Listad
o"
145 POSITION 3,17:? "7- Terminar"
147 POSITION 22,22:? " REGISTROS  ";NP
150 POSITION 1,20:? "ELIJA UN NUMERO :
 ";:GET #1,A:R$=CHR$(A)
160 IF R$<"1" OR R$>"7" THEN ? CHR$(25
3):GOSUB 6510:GOTO 150
170 TRAP 40000:ON VAL(R$) GOTO 500,100
0,2000,3000,3500,4000,5000
```

(continúa en página 29)



Avanzando

Esta columna pretende dar ciertos consejos y trucos para una programación más eficiente. Ella se dirige a personas que tienen conocimientos básicos de programación.

1. La estructura del programa:

Si bien el ATARI BASIC presenta problemas de estructuración un programador experimentado puede disminuirlos estableciendo un esqueleto en sus líneas de programación.

El rango de líneas de programa abarca desde la número 0 a la 32767.

Las rutinas generales de un programa tienen gran similitud, aunque los objetivos sean diferentes, como por ejemplo: rutinas de pantalla, ingreso de datos, validación de datos, lectura veloz del disco, movimiento de áreas de memoria con uso del IOCB, etc.

Una estructura clásica puede ser la siguiente asignación de líneas:

0-9 Titulación del programa
10 Dirigida con instrucción GOTO a la rutina de inicialización.
130-599 Rutina de ingreso por teclado.
600-605 Rutina de pausa.
1000-1999 Programa central específico.
2000-9999 Subrutinas de uso común.
10000-10999 Líneas de DATA.
11000-15999 Subrutinas gráficas.
16000-16999 Subrutinas de joystick, cursor, player/missil, set de caracteres, etc.
17000-19999 Rutinas de acceso al diskette.
20000-29999 Rutinas de sonido.
30000-31999 Rutinas de pantalla y efectos especiales.
32000-32767 Inicialización.

2. Uso de instrucción REM:

Los comentarios con REM se ubican en el primer número del área respectiva.

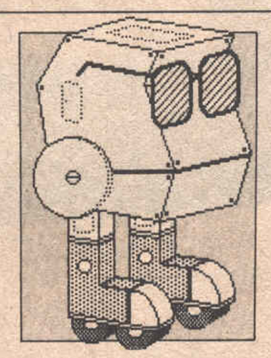
3. Valores numéricos constantes:

Un valor numérico cualquiera (ej. 10) requiere de 7 bytes de memoria. Al ser usado como un conjunto de variables requiere de 1 solo byte.

Supongamos que los valores 0, 1, 2, 4, 8, 16, 128, 256 se usen reiteradamente en un programa. El siguiente listado los asigna a las variables Z0, Z1, etc, hasta la Z256.

Para usar el valor 1 se utiliza entonces la variable Z1.

```
32200 REM VALORES PARA VARIABLES
32210 RESTORE 32220:READ Z0,Z1,Z2,Z3,Z
4,Z5,Z6,Z7,Z8,Z9,Z10,Z11,Z12,Z13,Z14,Z
15,Z16,Z255,Z256
32220 DATA 0,1,2,3,4,5,6,7,8,9,10,11,1
2,13,14,15,16,255,256
```



4. Uso de función USR:

MUNDOATARI, desde la creación de esta revista, ha estado consciente de las ventajas de la instrucción mencionada y la ha incorporado como columna permanente en sus páginas.

El uso exhaustivo de la función USR en la programación BASIC produce ventajas considerables en la rapidez del programa.

La búsqueda y optimización de estas rutinas debe llegar a constituir una serie de subrutinas con carácter de librería de uso permanente en la programación.

Para una librería de este tipo es necesario preparar un índice con los siguientes datos:

- Ubicación en memoria,
- Función específica,
- Parámetros de acceso,
- Significado de cada parámetro,
- Limitaciones de la rutina.

5. Uso de AUTORUN:

Para optimizar la presentación de su programa es necesario que el mismo contenga un programa inicial llamado AUTORUN.SYS, con instrucciones de carga de otro programa presente en el menú o programa central del diskette (ver al respecto en MUNDOATARI No. 17).

6. Menú de opciones:

La estructura de un programa de multitareas puede optimizarse en mayor grado con el control de un menú de acceso, de tal forma que en la partida se cargue el menú que permita direccionar hacia diferentes bloques precisos del sistema.

7. Protección del listado:

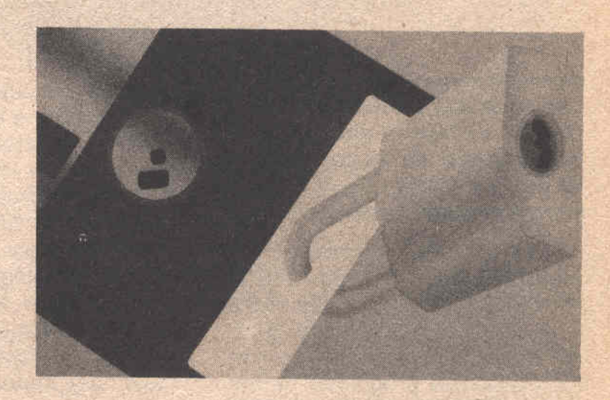
Los listados requieren de una protección para impedir que sean listados por el usuario.

Para ello puede utilizar la siguiente rutina localizada en la última línea de programación:

```
10 REM PROGRAMA PROTECCION LISTADO
32767 POKE PEEK(138)+256*PEEK(139)+2,0
:SAVE "D:NOMBRE.EXT":NEW
```

Para aplicar este listado debe tener una copia abierta del mismo para realizar posibles modificaciones futuras. Luego digite en modo directo:

GOTO línea de esta rutina bajo el nombre definido.



8. Protección del medio magnético:

- Casete: como consecuencia de la fácil duplicación de audio los sistemas de protección vía programación son poco seguros.
- Diskette: existen diferentes tipos de protección, de los cuales mencionaremos los más usados:

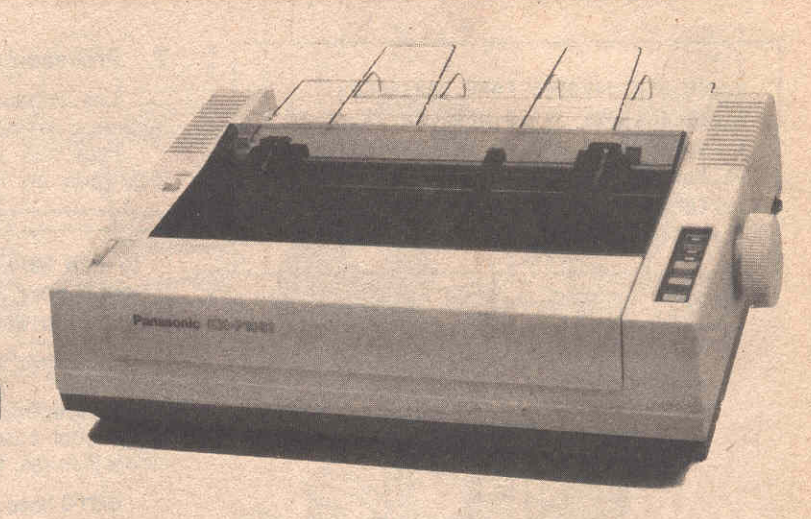
a) Modificación del directorio: permite crear un DOS particular con una ubicación determinada del directorio.

b) Creación de sectores malos: los que impiden la duplicación del disco mediante el DOS.

Hay en todo caso copiadores capaces de duplicar malos sectores.

c) Malos sectores con acceso del programa: esta técnica requiere crear malos sectores chequeados por el mismo programa. La duplicación requiere reproducir un sector defectuoso en las mismas condiciones.

d) Técnicas avanzadas: considerando el uso de formatos combinados en un mismo disco, lo cual impide la duplicación completa de la información. ●



Impresora 1080i

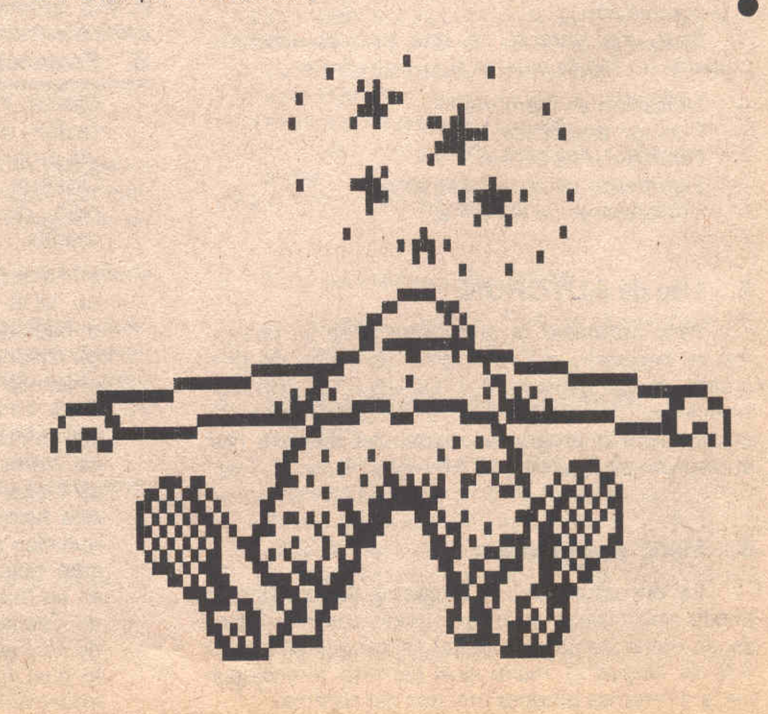
Las consultas de nuestros lectores nos inducen a emitir un informe acerca de las capacidades de esta impresora.

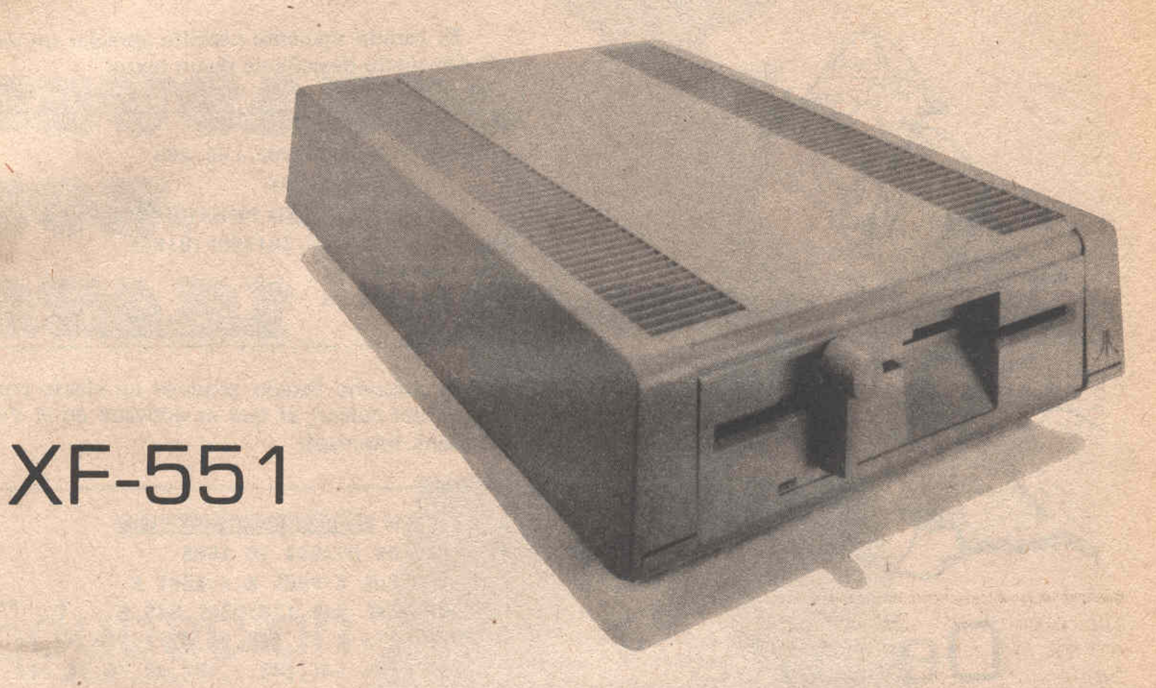
Los antiguos modelos de impresoras para computadores ATARI (1029, 1027, Seikosha y XM-801) provocan desencanto en los usuarios al utilizarlos con los excelentes programas de impresoras disponibles para el ATARI, tlaes como: Printshop o Newsroom.

Este hecho motivó aMUNDOATARI a seleccionar una impresora que por precio y potencialidad resuelva este problema. El modelo final quedó como la Panasonic 1080i, cuyo precio en nuestro Catálogo es inferior a la XM-801 y sus características son superiores en los siguientes aspectos:

1. Velocidad: 144 cps en lugar de los 100 de la XM-801.
2. Compatibilidad: dado que es Epson compatible, trabaja eficientemente los software ya mencionados.
3. Compatible con ST: utilizando el cable correspondiente.
4. Compatible PC: utilizando el cable correspondiente.
5. Características gráficas: calidad excelente como lo ilustran las muestras siguientes:
6. Set de caracteres: el modelo "i" contiene el set internacional de caracteres.
7. Manual: la impresora contiene un manual de 112 págs.
8. Cintas: de tipo cartridge, intercambiables y de bajo precio. También disponibles en el Catálogo MUNDOATARI.
9. Tipo de letra: imprime con letra de calidad, para trabajar como máquina de escribir.
10. Papel: usa hojas sueltas o formulario continuo.
11. Interfase: para el ATARI requiere un modelo que se proporciona en el Catálogo a un costo de $ 6.000 más IVA.

Por estas 10 razones hemos incluído este modelo de impresora para usted, amigo usuario, que busca lo mejor y a bajo precio.





XF-551

MUNDOATARI, consciente de la necesidad de los usuarios por mejorar sus configuraciones, inició el año pasado una estrategia de venta para colocar a su alcance este periférico.

Muchas interrogantes se plantea usted seguramente al momento de decidir la compra de un Drive (Unidad de Disco). ¿Cómo es este modelo en relación a la 1050 o a la 810?

El siguiente cuadro comparativo de las velocidades de acceso contesta un aspecto de dicha consulta.

Otro aspecto importante se refiere a la capacidad de almacenamiento:

- La 810 almacena 720 sectores de 128 bytes =90 Kb.
- La 1050 almacena 1024 sectores de 128 bytes = 128 Kb.
- La XF-551 almacena 1440 sects. de 256 bytes = 360 Kb.

El precio de la XF-551 en el Catálogo MUNDOATARI es de $ 75.990, lo que permite poner a su alcance este periférico, ya que es un 18 % inferior al precio normal de mercado.

El Sistema Operativo para formatear la capacidad total de la diskettera no ha sido producido masivamente por el fabricante. Pero MUNDOATARI, pensando en usted lo tiene en su poder para el análisis de su equipo. Como exclusividad proporcionamos una reproducción de la pantalla inicial de este nuevo Sistema Operativo.

Para aquellos usuarios que adquieran su XF-551 en MUNDOATARI existirá en el futuro la posibilidad de obtener sin costo una copia de este software.

MUNDOATARI proporciona otro Sistema Operativo que permite formatear igualmente los 360 Kb. mencionados, el Sparta DOS, en diskette, con manual en español. Además se entrega un diskette formateado con los 360 Kb. completo de juegos, lo que permite apreciar la magnitud de almacenamiento.

Considerando la necesidad de información técnica para este software MUNDOATARI ha estado publicando información al respecto en números anteriores.

Por estas razones, la elección de este periférico tendrá mayores ventajas si lo hace con la garantía y respaldo de MUNDOATARI.

VELOCIDAD COMPARATIVA DE DRIVES ATARI

Tipo de Drive	Formatear 720 K	Escribir 8 Kb		Leer 8 Kb	Escribir 32 Kb		Leer 32 Kb
		c/verif.	s/verif.		c/verif.	s/verif.	
810	0:35	0:27	0:14	0:08	1:44	0:50	0:31
1050	0:36	0:24	0:10	0:08	1:27	0:33	0:31
XF-551	0:26	0:22	0:09	0:07	1:23	0:31	0:29



De byte en byte

REGISTRO DE MODO DE CARACTERES

Localización 755 ($2F3)

El set de caracteres tiene una posibilidad de control de edición que es manejada por esta localización.

En el cuadro siguiente se puede observar los cambios operados por los valores entre 0 y 7 que se almacenan en la localización 755 con la instrucción POKE.

Valor decimal	0	1	2	3	4	5	6	7
CURSOR								
Transparente	x		x		x		x	
Opaco		x		x		x		x
Presente			x	x			x	x
Ausente	x	x			x	x		
CARACTERES								
Normales	x	x	x	x				
Invertidos					x	x	x	x

El cursor puede ser visible o invisible, transparente u opaco. Los textos pueden modificarse también en normal o inverso.

El listado siguiente permite apreciar un ejemplo del efecto destellante de un texto:

```
10 REM PROGRAMA LOC 755
15 ? CHR$(125)
20 PRINT "[illegible]"
30 POKE 755,INT(RND(0)*4)
40 A=1^1^1
50 GOTO 30
```

El próximo listado produce un efecto parpadeante del cursor, el que es activado en el Vertical Blank Interrupt:

```
5 REM CURSOR DESTELLANTE
10 FOR X=1664 TO 1680
15 READ A:POKE X,A:NEXT X
20 POKE 548,128:POKE 549,6
25 DATA 8,72,165,20,41,16,74,74,74
30 DATA 141,243,2,104,40,76,98,228
```

•

LOCALIZACION INICIAL DE MEMORIA LIBRE DISPONIBLE

Localizaciónes 743-744 ($ 2E7, $ 2E8)

Si encendemos el computador sin conectarlo con el Drive podemos averiguar el primer byte disponible en memoria RAM al digitar:

? PEEK(743) + 256 * PEEK(744)

El valor de retorno es 1792 ($ 700 hexadecimal).

Esta localización es útil para reservar espacio de memoria, desplazando el puntero hacia otra localización superior.

La siguiente rutina demuestra con la función USR la aplicación mencionada:

```
10 DIM ME$(24): PRO=700: REM número de
   bytes que se desea reservar **
15 HI=INT(PRO/256): LO=PRO-256*HI
20 FOR X=1 TO 24: READ A: ME$(X)=CHR$
   (A): NEXT X
30 ME$(6, 6)=CHR$(LO): ME$(14, 14)=CHR$
   (HI)
40 Z=USR(ADR(ME$))
50 DATA 24, 173, 231, 2, 105, 0, 141, 231, 2, 173,
   232, 2, 105
60 DATA 0, 141, 232, 2, 169, 0, 133, 8, 76, 0, 160
```



EDUCANDO CON ATARI

Educación

Estas vacaciones entregamos en la columna Educando con ATARI nada menos que dos programas para tipiar, que muestran aspectos interesantes de las posibilidades educativas del ATARI.

1. HISTORIA

Este programa contiene una pequeña biografía del prócer chileno Bernardo O'Higgins. Se destaca en él el uso de caracteres internacionales, la forma de edición en pantalla y el cambio de páginas.

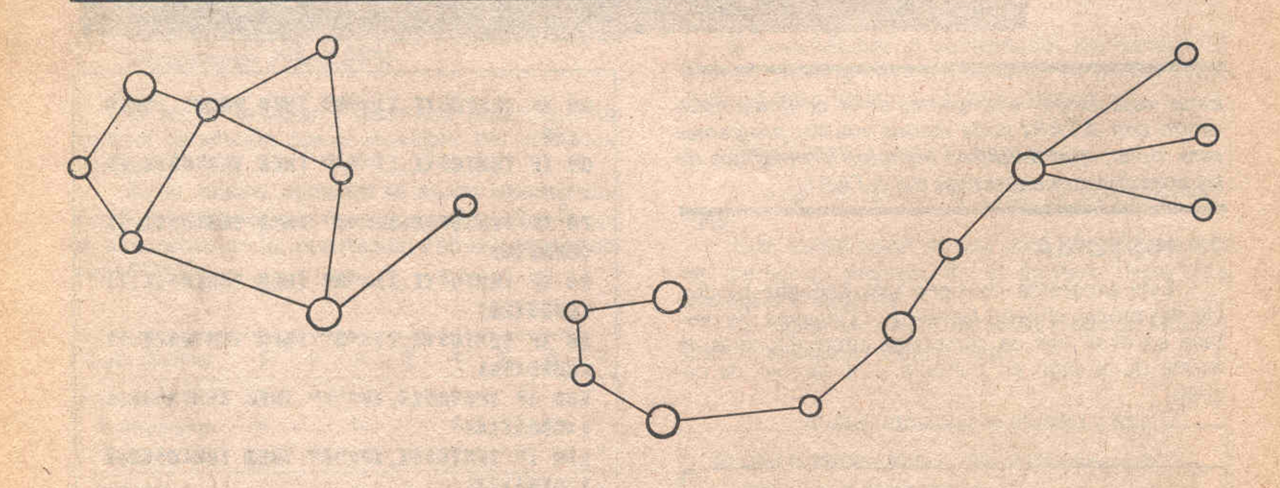
Digite y ejecute el siguiente listado:

```
10 DIM TEXTO$(600)
20 LT=1
25 REM ***  COLORES  ? **
28 GRAPHICS 0:POKE 756,204
30 READ TEXTO$
35 IF TEXTO$="" THEN GOTO 200
40 IF TEXTO$="FIN" THEN END
50 FOR I=1 TO LEN(TEXTO$)
```

```
60 IF TEXTO$(I,I)="@" THEN PRINT :GOTO
 150
65 IF TEXTO$(I,I)="/" THEN TEXTO$(I,I)
=","
70 IF TEXTO$(I,I)="!" THEN TEXTO$(I,I)
=CHR$(0)
80 IF TEXTO$(I,I)="#" THEN TEXTO$(I,I)
=CHR$(20)
90 IF TEXTO$(I,I)="$" THEN TEXTO$(I,I)
=CHR$(96)
100 IF TEXTO$(I,I)="%" THEN TEXTO$(I,I
)=CHR$(14)
110 IF TEXTO$(I,I)="&" THEN TEXTO$(I,I
)=CHR$(13)
120 IF TEXTO$(I,I)="<" THEN TEXTO$(I,I
)=CHR$(22)
130 PRINT TEXTO$(I,I);
140 FOR J=1 TO LT:NEXT J
150 NEXT I
160 GOTO 30
200 REM ** ESPERA [RETURN] **
210 POSITION 2,22:? "PRESIONE [RETURN]
```

```
PARA SEGUIR "
220 CLOSE #1:OPEN #1,4,0,"K:"
230 GET #1,A
240 IF A<>155 THEN 230
250 POSITION 2,3
260 FOR I=1 TO 21:? CHR$(156);:NEXT I
270 POSITION 2,2:GOTO 30
5000 DATA } BERNARDO O'HIGGINS@@
5005 DATA @ Don Bernardo O'Higgins er
a hijo del@gobernador de Chile don
 Ambrosio@O'Higgins y de do<a Isa
5010 DATA bel Riquelme.@Naci% en Chill
!n Viejo el 20 de agos-@to de 1778.@
5015 DATA @ Era un ni<o muy peque(o c
uando fue@alejado de su madre para se
r entrega-@do al cuidado de una fami
5020 DATA lia adinera-@da de Talca. Al
1$ le visit% su padre@cuando ten$a 9
a<os; esa fue la unica@oportunidad e
5025 DATA n que lo vio.@
5030 DATA @ Posteriormente fue envia
do a estu-@diar a Lima y cuando cumpl
i% 14 a<os@
5035 DATA fue embarcado por orden de d
on Ambro-@sio para que continuase su
formaci%n@en Inglaterra.@
```

```
5040 DATA
5045 DATA ++Separado de su madre y de
su tierra@natal/ sufr$o muchas amargur
as y pe-@nalidades.
5050 DATA @ En sus estudios tuvo #xit
o y siendo@ya un joven bien formado,
 conoci% a
5055 DATA @un personaje extraordinario
:Francisco@de Miranda. Este era un pat
riota ve-
5060 DATA @nezolano que viv$a en
Inglaterra@preocupado de impulsar la
 emancipa-
5065 DATA @ci%n de las colonias de Esp
a<a. Espe-@raba obtener la ayuda de
l gobierno
5070 DATA @ingl#s.@ O'Higgins se conv
irti% en admirador@de Miranda y compre
ndi% que deb$a co-
5075 DATA @laborar en sus planes. Jur%
 dedicarse@por completo a la causa d
e la inde-
5080 DATA @pendencia de Chile y de
 Am#rica y@desde entonces entreg% su v
ida a ese@ideal.
6000 DATA FIN
```



2. ESTRELLAS:

El programa consiste en un cuestionario de varias constelaciones estelares.

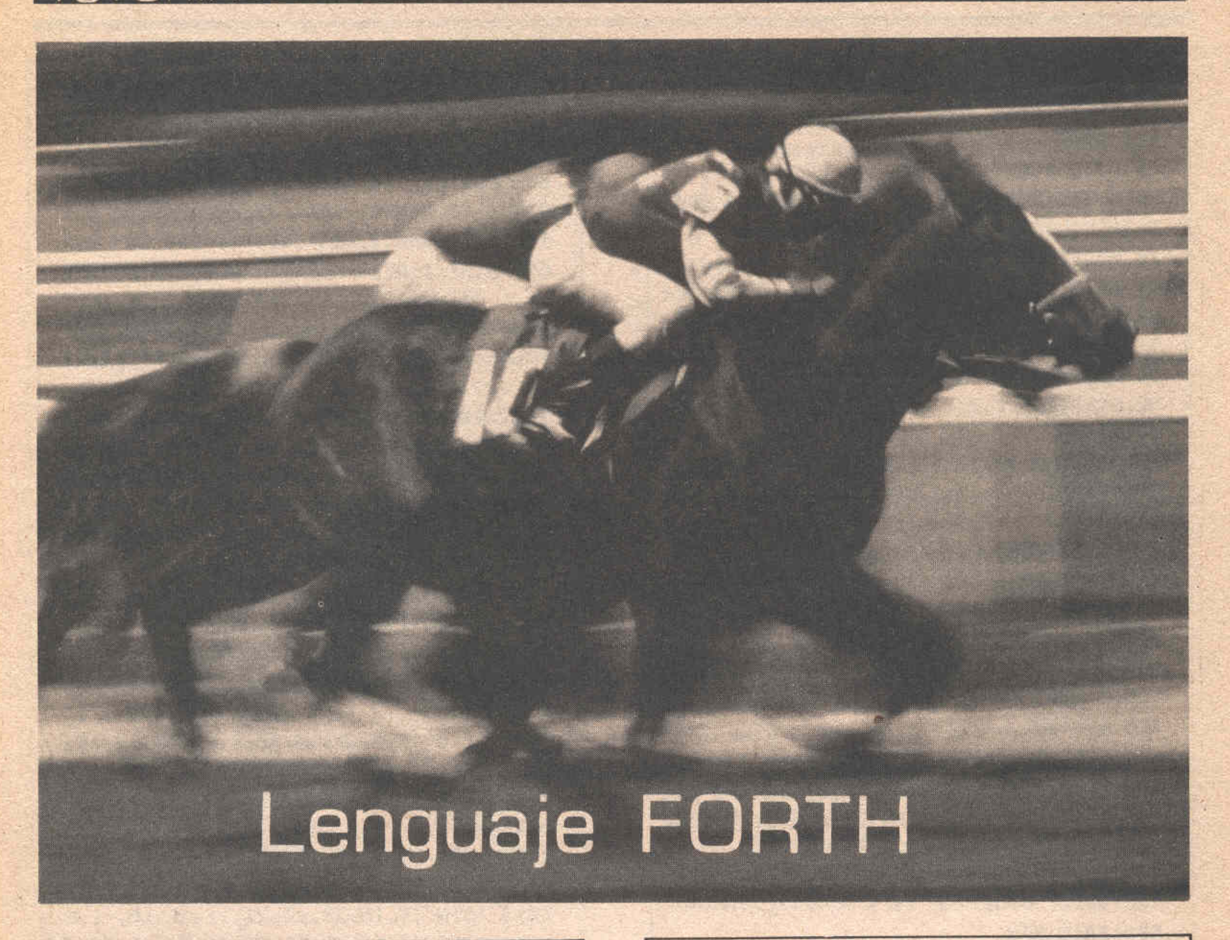
Presenta un esquema de una constelación y pregunta por su nombre. En caso de error presenta la respuesta acertada.

Digite el siguiente listado para comprobar sus conocimientos del tema:

```
10 REM ** ESTRELLAS **
20 DIM NOM$(15),RC$(20),RU$(20),X(10),
RCD$(20):GOSUB 4000
25 FOR I=1 TO 10:READ A:X(I)=A:NEXT I
27 DATA 85,76,72,85,60,85,25,25,50,100
30 ? "K":SETCOLOR 2,0,10:SETCOLOR 1,0,
1:SETCOLOR 4,0,6:POKE 752,0
```

```
40 POSITION 9,5:? "MAX. DE LETRAS=15"
50 POSITION 5,10:? "TU NOMBRE ES:";:IN
PUT NOM$
55 PTJE=0:NP=1
60 ? "K":SETCOLOR 2,0,1:SETCOLOR 1,0,1
0:SETCOLOR 4,0,1:POKE 752,1
70 POSITION 15,1:? "ESTRELLAS"
80 POSITION 2,3:? "PREGUNTA No ";NP
90 POSITION 2,5:? NOM$;" : ";PTJE;" PT
S."
94 IF PTJE>=MP THEN MP=PTJE
95 POSITION 2,7:? "MAX.PUNTUACION:";MP
;" PTS."
98 FOR I=0 TO 38:POSITION I,8:? CHR$(1
8):POSITION I,19:? CHR$(18):NEXT I:POK
E 752,0
100 L=INT(RND(0)*12):L=L+1000
110 RESTORE L
120 READ A,B
130 IF A=-1 THEN 145
140 POKE 752,1:POSITION A,B:? CHR$(20)
:GOTO 120
145 RESTORE L+2001
148 READ RC$,RCD$
150 POKE 752,0:POSITION 2,20:? "CUAL E
S EL NOMBRE DE LA CONSTELACION  QUE SE
 MUESTRA EN PANTALLA";:INPUT RU$
160 IF RU$=RC$ OR RU$=RCD$ THEN PTJE=P
TJE+10:IF PTJE>=MP THEN MP=PTJE:GOSUB
500:GOTO 170
165 GOSUB 300:GOSUB 600
170 NP=NP+1:IF NP=11 THEN 200
180 GOTO 60
200 ? "K":SETCOLOR 2,0,10:SETCOLOR 1,0
,1:SETCOLOR 4,0,6:POKE 752,0
210 POSITION 14,4:? "RESULTADOS"
220 POSITION 4,9:? NOM$;" TU PUNTAJE F
UE DE:";PTJE;" PTS."
225 POKE 752,1:FOR I=1 TO 38:POSITION
I,14:? CHR$(20):POSITION I,18:? CHR$(2
0):NEXT I
230 POSITION 2,16:? "DESEAS JUGAR DE N
UEVO 1=SI/2=NO ";:INPUT OP
240 IF OP=1 THEN 30
250 GRAPHICS 0:END
300 POKE 752,1:FOR I=18 TO 9 STEP -1:P
OSITION 0,I:? "
              ":K=K^1:NEXT I
310 POSITION 10,13:? "NOMBRE CORRECTO
 ";RC$:RETURN
500 ? "K":POKE 752,1:POSITION 5,9:? "!
!!!! C O R R E C T O!!!!!":L=57:H=45:P
=45:FOR A=1 TO 6:SOUND 0,P,10,14
510 FOR W=1 TO 180:NEXT W:P=L:L=H:H=P:
NEXT A:SOUND 0,0,0,0:RETURN
600 REM ** SONIDO DE ERROR **
610 FOR X=1 TO 5:SOUND 0,X(X),12,10:FO
R Y=1 TO X(X+5):NEXT Y:FOR Y=10 TO 0 S
TEP -1:SOUND 0,X(X),12,X(X+5)
620 NEXT Y:NEXT X:SOUND 0,0,0,0:RETURN

1000 DATA 9,11,11,14,9,17,13,13,13,14,
13,15,15,10,18,14,17,15,18,16,15,16,-1
,0
1001 DATA 13,12,17,9,18,13,21,13,20,17
,17,16,-1,0
1002 DATA 10,9,16,10,3,12,14,16,20,12,
-1,0
1003 DATA 11,16,11,13,14,11,17,15,20,1
4,21,12,21,11,-1,0
1004 DATA 11,13,15,10,21,10,13,15,17,1
6,21,15,18,13,-1,0
1005 DATA 11,16,14,11,20,9,17,16,-1,0
1006 DATA 15,10,21,12,15,13,10,18,23,1
4,-1,0
1007 DATA 18,9,20,13,22,9,10,18,17,14,
17,11,-1,0
1008 DATA 19,14,23,12,27,11,23,18,15,1
5,20,9,-1,0
1009 DATA 14,9,15,15,19,13,25,13,22,16
,-1,0
1010 DATA 15,15,19,10,25,11,22,16,-1,0
1011 DATA 18,10,22,16,23,14,24,11,25,9
,28,9,29,12,-1,0
3001 DATA ORION,ORION
3002 DATA LYRA,LYRA
3003 DATA CETUS,CETUS
3004 DATA OSA MAYOR,URSA MAJOR
3005 DATA AQUILA,AGUILA
3006 DATA PHOENIX,FENIX
3007 DATA CEPHEUS,CEPHEUS
3008 DATA LEPUS,LEPUS
3009 DATA CYGNUS,CISNE
3010 DATA CASSIOPEIA,CASIOPEA
3011 DATA DELPHINUS,DELFIN
3012 DATA OSA MENOR,URSA MINOR
4000 GRAPHICS 17:POSITION 5,7:? #6;"es
trellas":POKE 712,152:POKE 709,0
4010 FOR I=15 TO 6 STEP -0.2:POKE 709,
ABS(I-15)
4020 FOR W=I*4 TO I*2 STEP -1:SOUND 0,
W,10,I:NEXT W:NEXT I:SOUND 0,0,0,0
4030 POSITION 3,7:? #6;"          "
4040 FOR T=1 TO 200:NEXT T
4050 GRAPHICS 0:RETURN
```



Lenguaje FORTH

MUNDOATARI presenta en este número el uso del lenguaje FORTH, comparandolo con otros lenguajes corrientes, para que nuestros lectores puedan capacitarse en otro medio de comunicación con el computador.

Iniciaremos esta columna describiendo algunas ventajas del FORTH:

Para demostrar la capacidad de velocidad digite el siguiente listado, que se encuentra en BASIC y utiliza los timers internos (localizaciones 19 y 20) para medir el tiempo de ejecución:

```
140 GRAPHICS 24
150 REM
160 REM * ZERO SYSTEM JIFFY TIMERS
170 POKE 19,0:POKE 20,0
180 REM
190 REM * FIND ADDRESS OF SCREEN
200 SCREEN=PEEK(88)+256*PEEK(89)
210 REM
220 REM * SCREEN-FILL LOOP
230 FOR I=1 TO 32
240 FOR J=0 TO 239
250 POKE SCREEN+J,255
260 NEXT J
270 SCREEN=SCREEN+240
280 NEXT I
290 REM
300 REM * DETERMINE FILL TIME
310 JIFFIES=PEEK(20)+256*PEEK(19)
320 REM
330 REM * DISPLAY FILL TIME
340 GRAPHICS 0
350 PRINT JIFFIES;" JIFFIES":END
```

Comentario de líneas:

140 Modo gráfico 24.
170 Inicializa los timers (19 y 20).
200 Asigna a variable SCREEN la localización inicial de pantalla en la memoria.
230 Inicia ciclo para variable I (ciclo externo).
240 Incializa ciclo para variable J (ciclo interno).
250 Modifica contenido de una localización correspondiente a pantalla.
260 Instrucción NEXT para ciclo interno.
270 Incrementa la variable SCREEN en 240 (correspondiente al ciclo interno).
280 Instrucción NEXT para ciclo externo.
310 Asigna a la variable JIFFIES el tiempo actual calculado por los timers.
340 Modifica modo gráfico a 0.
350 Edita el valor de la variable JIFFIES.

Ejecute el programa y anote el tiempo empleado en su ejecución.

A continuación digite el próximo listado, también en BASIC, que permite a través de la función USR realizar la misma operación anterior:

```
160 REM * READ ML DATA INTO ML$
170 DIM ML$(45)
180 FOR I=1 TO 45
190 READ BYTE:ML$(I)=CHR$(BYTE)
200 NEXT I
210 REM
220 REM * SET UP GRAPHICS MODE
230 GRAPHICS 24
240 REM
250 REM * FILL SCREEN, GET TIMING
260 JIFFIES=USR(ADR(ML$))
270 REM
280 REM * DISPLAY TIMER READING
290 GRAPHICS 0
300 PRINT JIFFIES;" JIFFIES":END
310 REM
320 REM * DATA FOR ML ROUTINE
330 DATA 104,169,0,133,19,133,20,162,3
2,160,0,169,255,145,88,200,192,240,208
,249,24,165,88,105
340 DATA 240,133,88,165,89,105,0,133,8
9,202,208,229,165,19,133,213,165,20,13
3,212,96
```

Comentario de líneas:

170 Dimensiona la variable alfanumérica ML con 45 caracteres que contendrá la rutina en máquina.
180-200 Ciclo que permite leer los caracteres de las líneas 330-340 y localizarlos en la variable alfanumérica.
230 Incializa modo gráfico 24.
260 Utiliza la función USR en el formato:

JIFFIES = USR (ADR(ML$))

que ejecuta los ciclos de las líneas 230-280 del listado 1.
Es interesante destacar que el tiempo de demora retorna desde la función USR en la variable JIFFIES.
290 Modifica a modo gráfico 0.
300 Edita el valor de JIFFIES.
330-340 DATA de la función USR.

Ejecute el programa y compare los valores obtenidos.

Para nuestros amigos que siguen las columnas Directo al 6502 y USR proporcionamos paralelamente el listado en ASSEMBLER que corresponde a las líneas de DATA (330-340) del listado 2.

De esta manera podrán practicar y utilizar los conocimientos de esta nueva columna de continuidad en su revista MUNDOATARI.

Los siguientes listados deben ser listados y ejecutados con el FORTH. Ellos muestran la implementación de la rutina para llenar la pantalla en modo gráfico 24 con las correspondientes mediciones de tiempo:

```
Scr # 1
 ( BASIC/ML/FORTH BENCHMARK       )
 ( FORTH VERSION BY SALLY FORTH )

 0 VARIABLE READING

 88 CONSTANT SCREEN

 : SCREENFILL ( THE FILL LOOP )
   32 0 DO
   240 0 DO
   255 I SCREEN @ + C!
   LOOP
   240 SCREEN @ + SCREEN !
   LOOP ;

                                  -->

Scr # 2
 ( BENCHMARK CONTINUED            )

 : BENCHMARK
   24 GR.
   0 19 ! ( ZERO TIMERS )
   SCREENFILL
   256 19 C@ *  20 C@  +
   READING ! ( SAVE READING )
   0 GR.
   READING @ . ( SHOW READING )
   ." JIFFIES" CR ;
```

El siguiente cuadro comparativo indica los tiempos aproximados:

	Jiffies	Segundos
ATARI BASIC	4.316	71,9
FORTH	236	3,9
ASSEMBLER	5	0,1

Esta comparación deja en muy mal pie al ATARI BASIC. Sin embargo este tiempo puede ser optimizado y reducido a 54 segundos (3.243 jiffies) ▶

si realizamos las siguientes modificaciones:

```
100 GRAPHICS 24
110 POKE 19,0:POKE 20,0:SCREEN=PEEK(88
)+256*PEEK(89)
120 FOR I=SCREEN TO SCREEN+7680:POKE I
,255:NEXT I:JIFFIES=PEEK(20)+256*PEEK(
19)
130 GRAPHICS 0
140 PRINT JIFFIES;" JIFFIES":END
```

Comentario de líneas:

100 Define modo 24.
110 Inicializa localizaciones 19 y 20 y asigna a la variable SCREEN la primera localización de pantalla.
120 Inicia ciclo para variable I. Valor inicial desde el valor de SCREEN hasta 7680 localizaciones superiores y asigna en la variable JIFFIES el valor del tiempo empleado.
130 Modifica a modo gráfico 0.
140 Edita el tiempo en pantalla.

Ejecute el listado y compruebe lo afirmado.

A continuación presentamos un nuevo listado que entusiasmará a los seguidores del ATARI BASIC, ya que permite reducir el tiempo de operación a "12" jiffies sin usar la función USR.

El secreto consiste en aplicar la técnica de la propagación en una variable alfanumérica.

Digite el siguiente listado:

```
100 GRAPHICS 24
110 POKE 19,0:POKE 20,0
120 DIM S$(7680)
130 STRING=PEEK(140)+256*PEEK(141)
140 SCREEN=PEEK(88)+256*PEEK(89)
150 TABLE=PEEK(134)+256*PEEK(135)
160 OFFSET=SCREEN-STRING
170 HI=INT(OFFSET/256)
180 LO=OFFSET-(HI*256)
190 POKE TABLE+2,LO:POKE TABLE+3,HI
200 S$(1,1)=CHR$(255)
210 S$(7680,7680)=CHR$(255)
220 S$(2)=S$
230 JIFFIES=PEEK(20)+256*PEEK(19)
240 GRAPHICS 0
250 PRINT JIFFIES;" JIFFIES":END
```

Comentario de líneas:

100 Define modo gráfico 24.
110 Inicializa sistema de timers.
120 Dimensiona la variable alfanumérica S$ con 7.680 caracteres. Observe que este valor corresponde al total de posiciones de pantalla del modo 24.
130 Asigna en la variable STRING la localización que corresponde a la inicial de la variable alfanumérica S$.
140 Asigna en variable SCREEN la primera posición de pantalla.
150 Asigna en variable TABLE el valor en memoria para el puntero VVTP.
160 Asigna en variable OFFSET la diferencia entre STRING y SCREEN.
170 Busca el byte mayor de OFFSET.
180 Busca el byte menor de OFFSET.
190 Inserta los valores de posición de la variable STRING en la variable TABLE.
200-220 Rutina de propagación de variable S$.
230 Asigna a variable JIFFIES el tiempo.
240 Modifica a modo gráfico 0.
250 Edita el valor de JIFFIES.

Ejecute el listado y compruebe el tiempo.

Los nuevos usuarios se preguntarán tal vez: *¿Y qué pasa con el FORTH?*

El FORTH por su parte también puede optimizar su ejecución usando la palabra FILL como lo muestra el próximo listado. La ejecución en este caso demora sólo 7 jiffies.

```
Scr # 3
  ( FAST VERSION OF BENCHMARK    )
  ( BY SALLY FORTH               )

  0 VARIABLE READING

  88 CONSTANT SCREEN

  : BENCHMARK-2
   24 GR.
   0 19 !
   SCREEN @  7680 255 FILL
   256 19 C@ *  20 C@  +
   READING !
   0 GR.
   READING @ .
   ." JIFFIES" CR ;
```

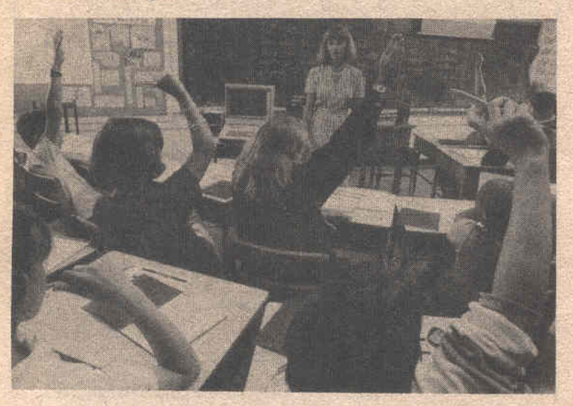
CONCLUSIONES:

El análisis de los listados anteriores y sus tiempos nos permite establecer lo siguiente:

1. Ningún lenguaje de alto nivel puede ser más rápido que el lenguaje de máquina.
2. El BASIC y el FORTH son lenguajes construidos en función de lenguaje de máquina, y su optimización depende de la práctica y habilidad del programador.
3. Es difícil optimizar la velocidad de BASIC sin recurrir a la función USR.

Finalizando esta introducción debemos recordar que la velocidad no es el único factor que determina la evaluación de un lenguaje, existen otros como: facilidad de uso, grado de eficiencia, etc. ●

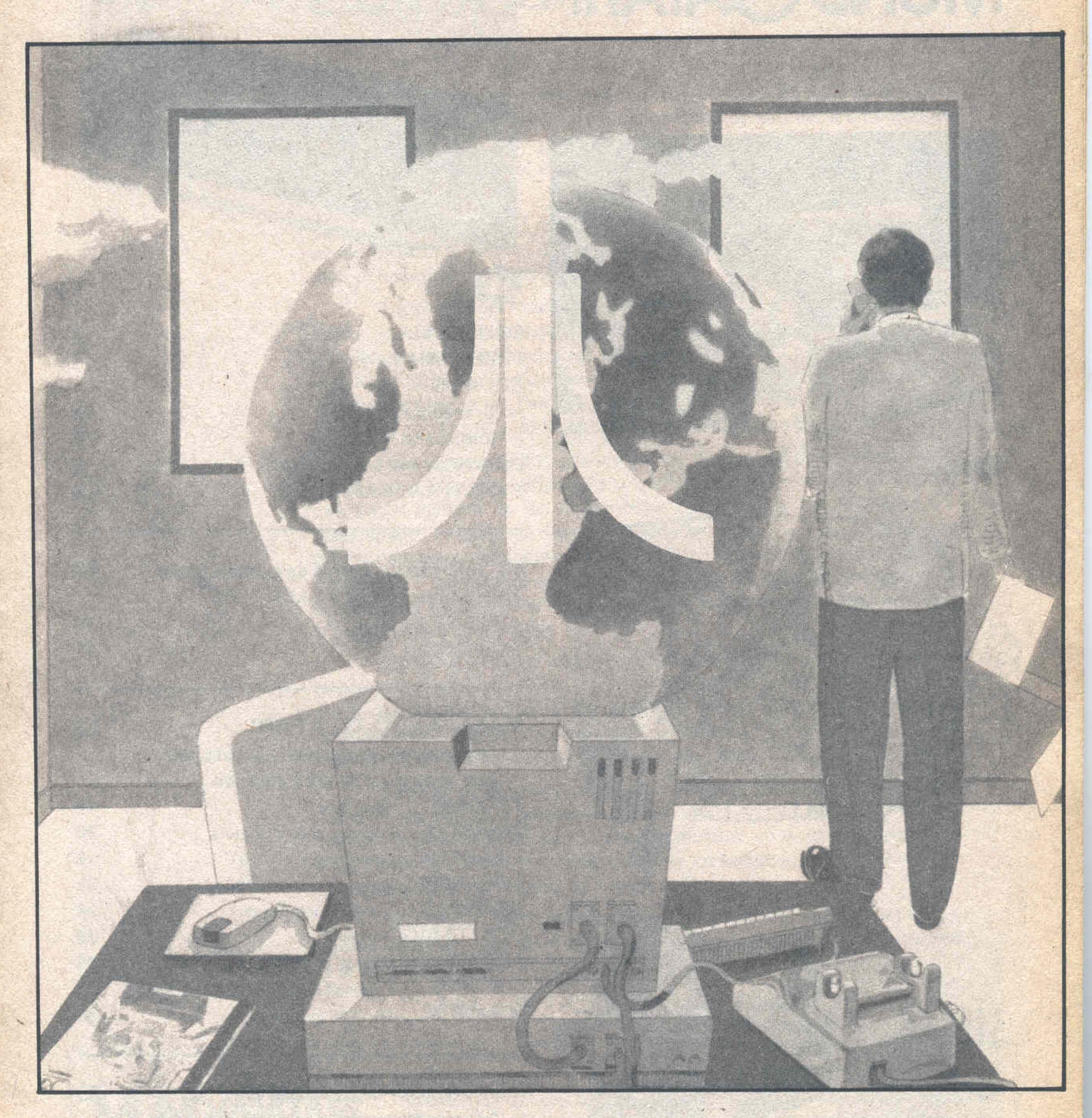
(☐ viene de página 17)



```
500 REM LISTADO
505 GRAPHICS 0:POKE 82,0:POKE 83,38:PO
KE 752,1:POKE 53774,64:POKE 16,64
506 IF NP=0 THEN 6585
510 POSITION 7,1:PRINT "DATOS GENERAL
ES"
515 POSITION 1,3:PRINT "ASIGNATURA: ":
POSITION 28,4:PRINT "CURSO: "
520 POSITION 0,5:? "Nro.  Nombres
                   Nota      "
528 E=1
530 FOR I=7 TO 18
531 POSITION 0,I:? "
                   "
532 IF E>NP THEN 560
535 B$=" ":B$=ARCH$(E*65-64,E*65)
536 IF E>1 THEN 542
540 POSITION 14,3:? B$(26,41):POSITION
 35,4:? B$(42,44)
542 IF E<10 THEN POSITION 2,I:? E
544 IF E>=10 THEN POSITION 1,I:? E
550 POSITION 5,I:? B$(1,25):POSITION 3
2,I:? B$(63,65)
558 E=E+1
560 NEXT I
570 POSITION 1,21:? "PRESIONE RETURN
 PARA SEGUIR ";:INPUT R$
585 IF E<=NP THEN 530
590 GOTO 105
1000 REM INGRESO DE DATOS
1005 B$=" ":B$(65)=" ":B$(2)=B$
1010 POKE 53774,64:POKE 16,64:POKE 752
,1:? CHR$(125):GOSUB 6000
1015 IF NP>50 THEN 1350
1020 POSITION 10,1:? "INGRESANDO DAT
OS":RU=1330
1025 POSITION 2,3:? "Asignatura:";:I
NPUT ASIG$:GOSUB 6530
1030 POSITION 25,4:? "Curso:";:INPUT
 CUR$:IF CUR$="**" THEN 1000
1035 GOSUB 6555:GOSUB 6570:NP=NP+1:N5=
NP:IF NP>50 THEN 1350
1040 POSITION 0,6:? N5;" Nombre:";:I
NPUT NOM$:IF NOM$="**" THEN GOSUB 6550
:NP=NP-1:GOTO 1030
1041 B$(1,25)=NOM$
1042 B$(26,41)=ASIG$
1043 B$(42,44)=CUR$
1048 GOSUB 6560:GOSUB 6510
1050 POSITION 2,8:? "NOTAS":POSITION
 1,10:? " N1   N2   N3   N4   N5   N6
   PROM. "
1065 GOSUB 6540
1070 REM CALCULO DE NOTAS
1072 POSITION 1,22:? "Tipee 0 return
 para terminar"
1080 I=0:P=0:PROM=0:W=12:PRM=0:Y=45
1090 FOR Z=2 TO 28 STEP 5:I=I+1
1100 POSITION 1,20:? "INGRESE NOTA: ";
I;" ";:INPUT NX$:IF NX$="" THEN ? CHR$
(253):NP=NP-1:GOTO 105
1102 NX=VAL(NX$)
1103 GOSUB 6510:IF NX<-1 OR NX>7 THEN
1100
1105 IF NX=0 AND I=1 THEN 1140
1110 N(I)=NX:IF NX=0 THEN I=I-1:Z=32:G
OTO 1140
1120 P=P+N(I)
1125 B$(Y,Y+2)=STR$(N(I)):Y=Y+3
1130 POSITION Z,W:? N(I):NEXT Z
1140 PRM=P/I:PROM=INT(10^1*PRM+0.5)/10
^1
1142 IF I=1 THEN A=7
1143 IF I=2 THEN A=12
1144 IF I=3 THEN A=17
1145 IF I=4 THEN A=22
1146 IF I=5 THEN A=27
1147 IF I=6 THEN 1150
1148 FOR P=A TO 37:POSITION P,W:? " ":
NEXT P
1150 POSITION Z+2,W:? PROM:GOSUB 6510
1160 GOSUB 6510:POSITION 1,20:? "CORRI
GE S/N ";:GET #1,A:R$=CHR$(A)
1165 IF R$="S" THEN 1070:IF R$="N" THE
N 1200:? CHR$(253):GOTO 1160
1200 REM ARCHIVO
1260 B$(63,65)=STR$(PROM)
1320 ARCH$(N5*65-64,N5*65)=B$
1325 B$=" ":B$(65)=" ":B$(2)=B$:GOSUB
6510:GOTO RU
1327 REM
```

```
1330 POSITION 1,20:? "INGRESA OTRA S/N
 ";:GET #1,A:R$=CHR$(A):GOSUB 6510
1335 IF R$="S" THEN 1035:IF R$="N" THE
N 1345:? CHR$(253):GOTO 1330
1345 TRAP 1330:GOTO 105
1348 REM
1350 REM ARCHIVO LLENO
1360 ? CHR$(125):POSITION 10,10:? CHR$
(253);"ARCHIVO LLENO "
1365 FOR T=1 TO 400:NEXT T:GOTO 105
2000 REM MODIFICA PERSONAS
2001 B$=" ":B$(65)=" ":B$(2)=B$
2002 GRAPHICS 0:POKE 53774,64:POKE 16,
64:POKE 752,1:IF NP=0 THEN 6585
2004 POSITION 22,22:? " Registros  ";
NP
2005 TRAP 2002:POSITION 1,20:? "QUE NU
MERO MODIFICA ";:INPUT NUM
2007 TRAP 40000:FOR X=1 TO NP
2010 IF NUM=X THEN 2030
2012 NEXT X
2015 GOSUB 6510
2020 POSITION 1,20:? CHR$(253);" PERSO
NA NO EXISTE EN LISTA"
2025 FOR T=1 TO 300:NEXT T:GOTO 105
2030 B$=" ":B$=ARCH$(NUM*65-64,NUM*65)
2035 NS=NUM
2050 ? CHR$(125):POSITION 10,1:? "MOD
IFICANDO PERSONA "
2060 POSITION 2,3:? " Asignatura : ";B
$(26,41):POSITION 25,4:? " Curso :";B$
(42,44):W=12
2065 POSITION 0,6:? X;" Nombre : ";B$(
1,25)
2067 POSITION 2,8:? " NOTAS ":POSITION
 1,10:? "  N1   N2   N3   N4   N5   N6
   PROM. "
2070 POSITION 2,W:? B$(45,47):POSITION
 7,W:? B$(48,50):POSITION 12,W:? B$(51
,53):POSITION 17,W:? B$(54,56)
2075 POSITION 22,W:? B$(57,59):POSITIO
N 27,W:? B$(60,62):POSITION 34,W:? B$(
63,65)
2100 GOSUB 6510:POSITION 1,20:? "MODIF
ICA S/N ";:GET #1,A:R$=CHR$(A)
2105 IF R$="N" THEN 2150:? CHR$(253):G
OTO 2100
2108 IF R$="S" THEN 6620
2110 RU=2150:GOSUB 6510:POSITION 1,20:
? "MODIFICA NOMBRE Y NOTAS[1] o NOTAS[
2]"
2120 POSITION 1,21:? "DIGITE NUMERO OP
CION ";:GET #1,A:R$=CHR$(A)
2125 IF VAL(R$)<>1 THEN 2140
2130 GOSUB 6580:GOSUB 6510:GOSUB 6520:
GOSUB 6600:GOSUB 6610:GOTO 1040
2140 GOSUB 6510:GOSUB 6520:GOSUB 6610:
GOTO 1070
2150 GOSUB 6510:GOSUB 6520:POSITION 1,
20:? "MODIFICA OTRA S/N ";:GET #1,A
2160 R$=CHR$(A):IF R$="N" THEN 105
2170 IF R$="S" THEN 2000:? CHR$(253):G
OTO 2150
3000 REM ELIMINA PERSONA
3002 GRAPHICS 0:POKE 53774,64:POKE 16,
64:POKE 752,1
3004 IF NP=0 THEN 6585
3005 POSITION 4,5:? " ELIMINA  PERSONA
 "
3010 POSITION 2,20:? "DEME NUMERO ";:I
NPUT NUM
3015 FOR X=1 TO NP:IF NUM=X THEN 3030:
NEXT X
3017 IF NUM=X THEN 3030
3020 NEXT X
3025 GOSUB 6510:POSITION 2,20:? " PERS
ONA NO EXISTE ":? CHR$(253):FOR T=1 TO
 400:NEXT T:GOTO 105
3030 REM BORRA
3035 PI=1+65*(X-1):PF=PI+64
3040 IF LEN(ARCH$)=65 THEN ARCH$=" ":G
OTO 3060
3044 TRAP 40000
3045 IF PI=1 THEN ARCH$=ARCH$(PF+1):GO
TO 3060
3050 IF NUM=NP THEN ARCH$=ARCH$(1,PF-6
5):GOTO 3060
3055 ARCH$(LEN(ARCH$(1,PI-1))+1)=ARCH$
(PF+1)
3060 GOSUB 6510:NP=NP-1:POSITION 3,20:
? CHR$(253);" PERSONA  ELIMINADA  ":FO
R T=1 TO 400:NEXT T
3070 POSITION 1,20:? "DESEA ELIMINAR O
TRA S/N ";:GET #1,A:R$=CHR$(A)
3080 IF R$="S" THEN 3000
3090 GOTO 105
3500 REM IMPRESION
3501 GRAPHICS 0:POKE 82,0:POKE 83,38:P
OKE 53774,64:POKE 16,64:POKE 752,1
3502 IF NP=0 THEN 6585:EL=1
3504 POSITION 8,5:? "  A D V E R T E N
 C I A ":? :? :? "  ANTES DE ESCOGER
 OPCION  DEBE ":?
3506 ? "  ENCENDER LA IMPRESORA."
```

MUNDO ATARI®



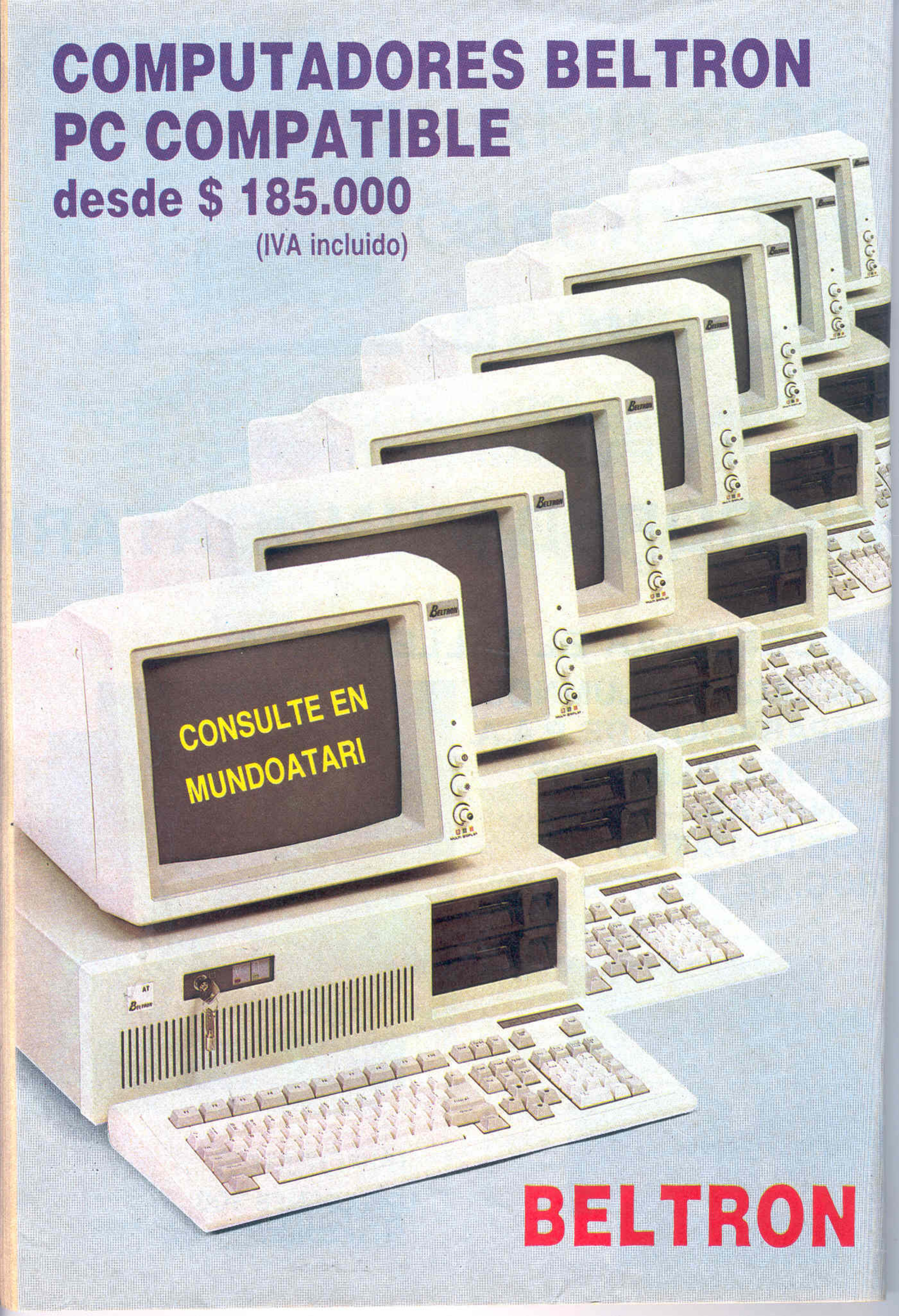
REEDICION DE REGALO

REVISTA Nº 0

```
5045 POSITION 3,15:? "PRESIONE RETURN
CUANDO ESTE LISTO ";:INPUT R$
5050 POSITION 10,21:? " GRABANDO ARC
HIVO ":CLOSE #2
5060 OPEN #2,8,0,"C:"
5070 FOR E=1 TO NP
5080 B$=" ":B$=ARCH$(E*65-64,E*65):PRI
NT #2;B$
5090 NEXT E:CLOSE #2
5092 ARCH$=" ":ARCH$(6500)=" ":ARCH$(2
)=ARCH$:POP :NP=0:GOSUB 5200
5095 POP :GOTO 105
5200 REM GENERA O LEE ARCHIVO EN CASSE
TTE
5210 ? CHR$(125):CLOSE #1:OPEN #1,4,0,
"K:"
5220 POSITION 13,2:? " O P C I O N E S
 "
5230 POSITION 2,4:? " L  LEER ARCHIVO
 DESDE CASSETTE"
5240 POSITION 2,6:? " C  CREAR ARCHIV
O EN CASSETTE"
5250 POSITION 2,8:? " M  MENU PRINCIP
AL"
5260 POSITION 2,20:? "ESCOJA UNA LETRA
 : ";:GET #1,A:R$=CHR$(A)
5270 IF R$="L" THEN 5500
5280 IF R$="C" THEN POP :GOTO 1000
5290 IF R$="M" THEN RETURN
5300 GOTO 5200
5500 REM LEE ARCHIVO DESDE CASSETTE
5510 ? CHR$(125)
5520 POSITION 10,2:? " LECTURA DE ARC
HIVO "
5530 POSITION 6,5:? "PASOS A SEGUIR"
5540 POSITION 1,7:? "1- Rebobine Cinta
"
5550 POSITION 1,9:? "2- Presione Tecla
 PLAY de la Grabadora"
5560 POSITION 1,11:? "3- Presione 2 Ve
ces Tecla RETURN"
5570 POSITION 1,16:? "PRESIONE RETURN
CUANDO ESTE LISTO ";:INPUT R$
5580 POSITION 10,21:? " LEYENDO ARCH
IVO ":CLOSE #2
5590 OPEN #2,4,0,"C:":ARCH$=" ":NP=0
5595 TRAP 5620
5600 B$=" ":INPUT #2;B$:NP=NP+1
5610 ARCH$(NP*65-64,NP*65)=B$:GOTO 560
0
5620 TRAP 40000:IF PEEK(195)=136 THEN
CLOSE #2:GOTO 105
```

```
5630 POP
5700 REM RUTINA DE ERROR
5710 ? CHR$(125):? CHR$(253)
5720 POSITION 8,2:? " A D V E R T E N
C I A "
5730 POSITION 3,7:? "LO LAMENTO, NO HA
 CARGADO EL PRO-     GRAMA COMPLETO.
REPITA LA ACCION"
5735 POSITION 3,9:? "O DEBERA CREAR AR
CHIVO DE LO CON-     TRARIO NO PODRA S
EGUIR ADELANTE"
5740 POSITION 2,20:? "PRESIONE RETURN
PARA CONTINUAR ";:INPUT R$:GOTO 100
6000 REM LIMPIA VARIABLES
6010 R$=" ":ASIG$=" ":CUR$=" ":NOM$="
"
6500 REM LIMPIA LINEAS
6510 POSITION 0,20:? "
                    ":RETURN
6520 POSITION 0,21:? "
                    ":RETURN
6530 POSITION 1,22:? "Tipee ** para Co
rregir ":RETURN
6540 POSITION 0,22:? "
                    ":RETURN
6550 POSITION 34,4:? "    ":RETURN
6555 POSITION 0,6:? "
                  ":RETURN
6560 GOSUB 6510:POSITION 1,20:? "CORRI
GE S/N ";:INPUT R$:IF R$="N" THEN 6540
6565 IF R$="S" THEN POSITION 8,5:? "
                   ":GOTO 1040
6570 POSITION 0,12:? "
                  ":RETURN
6580 POSITION 12,6:? "
       ":RETURN
6585 ? CHR$(125);CHR$(253)
6586 POSITION 1,6:? " Lo siento, lo qu
e Ud. solicita no se   puede realizar.
 No tengo registros    en MEMORIA"
6587 ? :? "    Debe  Crear  el  Arch
ivo  ":FOR T=1 TO 400:NEXT T:GOTO 100
6600 B$(1)=" ":B$(25)=" ":B$(2)=B$:RET
URN
6610 FOR ZZ=45 TO 65:B$(ZZ,ZZ)=" ":NEX
T ZZ:RETURN
6620 GOSUB 6510:POSITION 1,20:? "INGRE
SE CLAVE DE ACCESO : ";:INPUT R$
6622 IF R$="GCQ" THEN 2110
6624 ? CHR$(125);CHR$(253):POSITION 10
,5:? " RECHAZADO ":? :? :? "      SO
LO PUEDE VER DATOS ":GOTO 2150
```


MUNDOATARI®
MUNDOATARI®
TE INFORMA
SU NUEVA
DIRECCION...
Av. LOS LEONES
308


COMPUTADORES BELTRON
PC COMPATIBLE
desde $ 185.000
(IVA incluido)
CONSULTE EN
MUNDOATARI
BELTRON



Tome
Coca-Cola
Coke
MARCAS REG.

AHORA EN MUNDOATARI IMPRESORAS PANASONIC 1080i - 144 cps.

$ 85.990 (IVA incluido)

Panasonic KX-P1081

CREDITOS



Los Leones 308
[P]rovidencia – Santiago